

Oracle® Hyperion Financial Data Quality Management, Enterprise Edition

Administrator's Guide



Release 11.1.2.4.220
E73556-07
December 2017

ORACLE®

Oracle Hyperion Financial Data Quality Management, Enterprise Edition Administrator's Guide, Release 11.1.2.4.220

E73556-07

Copyright © 2009, 2017, Oracle and/or its affiliates. All rights reserved.

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 installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. 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 Xeon 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, Opteron, the AMD logo, and the AMD Opteron 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 Using FDMEE

Supported Source Systems	1-1
File-Based Data Loads	1-2
Supported EPM System Target Applications	1-2
Key FDMEE Benefits	1-3
FDMEE Application Folder Architecture	1-3
FDMEE Architecture	1-6
Drilling Through	1-8
Drilling Through to the FDMEE Landing Page	1-9
Configuring Oracle Data Integrator With FDMEE	1-10
Set up the Data Server Based on the Enterprise Resource Planning (ERP) Source System	1-10
Setting up Physical Schemas	1-11
Setting up the ODI Context Code	1-12
Setting up ODI to Integrate with JD Edwards	1-12
Security	1-17
Integration Process Overview	1-19
Extracting General Ledger Data	1-19
Writing Back Data from EPM Applications	1-19
Integrating Metadata	1-20
Understanding General Ledger Integration	1-20
Requirements	1-21
Required Dimensions	1-22
Member Properties Sourced from the Enterprise Resource Planning (ERP) System	1-22
How Dimensions Are Processed	1-26
How Languages Are Processed	1-27

How Currencies Are Processed	1-28
Defining Metadata Rules	1-29
Loading Source System Hierarchies into EPM Dimensions	1-36
Extracting Human Resource Data	1-36
Navigating FDMEE	1-37
Toolbars	1-37
Help Menu	1-37
Task Pane Options	1-37
Workflow Tasks	1-37
Setup Tasks	1-38
Working with Data in Grids	1-38
FDMEE User Interface Elements	1-39
Advanced Search Options	1-40
Using the POV Bar	1-41
Selecting the Location POV	1-41
Setting the Period POV	1-42
Setting the Category POV	1-42
Locking and Unlocking a POV	1-43
Administration Tasks	1-44
Predefining a List of Profiles	1-44
Setting System-Level Profiles	1-44
Setting Application-Level Profiles	1-52
Setting User Level Profiles	1-57
Setting Security Options	1-59
Setting up Source Systems	1-63
Registering Enterprise Resource Planning (ERP) Source Systems	1-64
Registering File-Based Source Systems	1-66
Deleting Registered Source Systems	1-66
Editing Registered Source System Details	1-67
Adding File-Based Data Load Definitions	1-67
Working with Source Adapters	1-67
Using a Prepackaged Integration for SAP	1-68
Adding Source Adapter Definitions	1-69
Defining Source Adapter General Information	1-69
Defining Source Adapter Detail	1-72
Registering Target Applications	1-78
Creating a Custom Target Application	1-80
Adding Lookup Dimensions	1-83
Defining Application Dimension Details	1-83
Defining Application Options for Essbase and Planning	1-84
Defining Application Options for Profitability and Cost Management (HPCM)	1-92

Defining Application Options for Public Sector Planning and Budgeting	1-95
Defining Application Options for Financial Management	1-96
Deleting Registered Target Applications	1-106
Reusing Target Application Names Multiple Times	1-107
Selecting Source Accounting Entities	1-107
Assigning General Ledger Responsibility	1-108
Working with Source Accounting Entity Groups	1-109
Loading Excel Data	1-110
Downloading to Excel	1-111
Uploading from Excel	1-112

2 Integrating Tasks

Working with Import Formats	2-1
Defining the Import Format	2-2
Viewing Import Format Information	2-2
Adding Import Formats	2-2
Deleting an Import Format	2-5
Querying by Example	2-5
Defining Import Format Mappings	2-6
Defining the Import Format Mappings	2-7
Concatenating Source Dimensions for Enterprise Resource Planning (ERP) Segments	2-8
Adding Import Expressions	2-9
Import Expression Types	2-10
Processing Order	2-13
Defining Import Formats for File-Based Mappings	2-13
Concatenating Source Dimensions for a File-Based Source	2-16
Using the Import Format Builder	2-17
All Data Types Data Loads	2-19
All Data Types Data Load Process Description	2-19
Setting the All Data Types Load Method	2-20
Setting the All Data Types Load Method	2-21
Setting the Import Format Data Types	2-22
Setting the Import Format for Multi-Column Data Types	2-23
Loading Incremental Data using the LINEITEM Flag to an EPM Application	2-28
Loading Multi-Column Numeric Data	2-31
Loading Periods As a Column from the Data File	2-34
Write-Back Mappings	2-35
Defining Import Formats for Data Synchronization Between EPM Applications	2-36
Defining Import Formats for Data Synchronization Between Enterprise Resource Planning (ERP) and EPM Applications	2-37

Defining the Write-Back Mappings (E-Business Suite and PeopleSoft only)	2-39
Defining Locations	2-39
Defining Period Mappings	2-42
Global Mappings	2-45
Application Mappings	2-45
Source Mappings	2-46
Defining Category Mappings	2-48
Global Mappings	2-49
Application Mappings	2-49

3 Integrating Data

Loading Data	3-1
Creating Member Mappings	3-1
Creating Mappings Using the Explicit Method	3-3
Creating Mappings Using the Between Method	3-4
Creating Mappings Using the In Method	3-4
Creating Mappings Using the Multi-Dimension Method	3-5
Using Special Characters in Multi-Dimensional Mapping	3-6
Creating Mappings Using the Like Method	3-7
Using Special Characters in the Source Value Expression for Like Mappings	3-7
Automap Wildcarding	3-10
Using Special Characters in the Target Value Expression	3-12
Format Mask Mapping for Target Values	3-13
Ignoring Member Mappings	3-16
Importing Member Mappings	3-17
Downloading an Excel Template (Mapping Template)	3-19
Importing Excel Mappings	3-20
Exporting Member Mappings	3-22
Deleting Member Mappings	3-22
Restoring Member Mappings	3-23
Defining Data Load Rules to Extract Data	3-23
Defining Data Load Rule Details	3-23
Defining Data Load Rule Details for a File-Based Source System	3-42
Defining Source Parameters for Planning and Essbase	3-44
Defining Source Parameters for Financial Management	3-45
Managing Data Load Rules	3-47
Editing Data Load Rules	3-47
Running Data Load Rules	3-47
Scheduling Data Load Rules	3-52
Checking the Data Load Rule Status	3-52

Deleting Data Load Rules	3-52
Working with Target Options	3-53
Creating Custom Options	3-53
Loading Exchange Rates to Financial Management	3-53
Using Drilling Through	3-55
Creating the Drill Region	3-55
Drill Through Components	3-56
Adding the Server Component for the Drill Through URL	3-56
Adding the Detail Component for the Drill Through URL	3-57
Viewing the Drill Through Results	3-59
Integrating with the EPM Cloud	3-62
Configuring the Secure Sockets Layer (SSL) Certificate	3-63
Pre-requisites	3-65
Setting up the EPM Cloud Deployment	3-66
Loading Data to the EPM Cloud or On-Premise Application	3-66
Exporting the EPM Cloud Application	3-66
Using the Data Load Workbench	3-66
Workflow Grid	3-67
Processing Data	3-67
Using the Workbench Data Grid	3-71
Viewing Process Details	3-79
Integrating with the Oracle Financials Cloud	3-80
Integration Process Description	3-81
Configuring a Source Connection	3-82
Working with Import Formats	3-85
Defining Locations	3-86
Defining Category Mappings	3-88
Data Load Mapping	3-88
Adding Data Load Rules	3-90
Processing Oracle General Ledger Adjustment Periods	3-91
Adding Filters for Data Load Rules	3-93
Drilling Through to the Oracle Financials Cloud	3-95
Writing Back to the Oracle Financials Cloud	3-95
Extracting the Integrated EPM Cloud and Oracle General Ledger Data to a Flat File	3-99
Using Excel Trial Balance Files to Import Data	3-102
Text Trial Balance Files Versus Excel Trial Balance Files	3-103
Downloading an Excel Trial Balance Template	3-103
Defining Excel Trial Balance Templates	3-103
Adding a Multiple Period Data Load Using Excel	3-104
Importing Excel Mapping	3-105

Using Journal Templates to Import Data	3-106
Additional Considerations for Loading Journal Templates	3-106
Integrating Financial Management Journals	3-106
Downloading a Journal Template	3-108
Defining Journal Templates	3-108
Processing Journals	3-111
Loading Data Using a Universal Data Adapter	3-114
Defining a Universal Data Adapter in Oracle Database Integrator (ODI)	3-114
Configuring SAP HANA	3-117
Working with a Universal Data Adapter in FDMEE	3-118
Universal Data Adapter Source Systems and Import Formats	3-124
Universal Data Adapter Source Systems and Locations	3-125
Data Load Mapping for Universal Data Adapter Related Sources	3-125
Data Load Rules for Universal Data Adapter Related Source	3-125
Integrating Data Relationship Management with FDMEE	3-128
Setting up the Integration	3-129
Data Relationship Management and FDMEE Locations	3-130
Data Relationship Management and FDMEE Metadata Rules	3-131
Importing Data Load Mappings from Data Relationship Management	3-132
Financial Close Management Integration with FDMEE	3-133
Data Load, Synchronization and Write Back	3-133
Overview	3-134
Synchronizing and Writing Back Data	3-134
Data Synchronization	3-134
Write-Back	3-141
Loading Human Resources Data	3-152
Understanding Human Resource Data Integration	3-152
Requirements	3-153
Running Processes in PeopleSoft Human Capital Management	3-153
Defining the Data Load Settings in Planning	3-153
Public Sector Planning and Budgeting Dimensions	3-154
Smart Lists	3-154
Using Loaded Data in Public Sector Planning and Budgeting	3-155
Defining Human Resource Data Load Rules	3-155
Defining the Rule Details	3-155
Creating Rule Line Definitions by Copying	3-156
Creating Rule Line Definitions by Using a Template Mapping	3-157
Creating Rule Line Definitions	3-157
Creating Mapping Definitions	3-158
Managing Human Resource Data Load Rules	3-159
Editing Human Resource Data Load Rules	3-159

Running Human Resource Data Load Rules	3-159
Deleting Human Resource Data Load Rules	3-161
Checking the Human Resource Data Load Rule Status	3-161

4 Logic Accounts

Overview of Logic Accounts	4-1
Creating a Logic Group	4-1
Creating Accounts In A Simple Logic Group	4-2
Logic Group Fields	4-2
Operator and Value/Expression	4-3
Expressions and Functions	4-4
Value/Expression	4-7
Seq	4-7
Export	4-7
Creating Summarized Logic Accounts	4-7
Creating Complex Logic Accounts	4-8
Complex Logic Example 1: CashTx	4-9
Sample Imported Values	4-10
Sample Imported Account Names	4-10
Final Result	4-10
Complex Logic Example 2: CashTx	4-10
Sample Imported Values	4-11
Logic Members	4-11
Final Result	4-12

5 Check Rules

Overview of Check Rules	5-1
Creating Check Rule Groups	5-2
Creating a Check Rule	5-2
Rule Logic	5-4
Using the Rule Logic Editor to Create Check Rules	5-4
Adding Rule Logic	5-6
Adding a Rule Logic Statement as Free Form Text	5-11
Testing Check Rule Expressions	5-12
Running Check Reports for Profitability and Cost Management BSO Essbase Cube	5-13
Creating Check Entity Groups	5-14

6 Batch Processing

Working with Batch Definitions	6-1
Adding a Batch Group	6-7
Executing Batches	6-8
Using Open Batches	6-8
Name Format for Open Batch Files	6-9
Naming Conventions	6-9
Open Batch Import and Export Load Methods	6-10
Creating Open Batches	6-11
Creating an Open Batch to Run an Integration with E-Business Suite	6-13
Creating Open Batches for Multiple Periods	6-13
Scheduling Jobs	6-15
Canceling a Scheduled Job	6-18
Working with Batch Scripts	6-18
Using Password Encryption	6-19
Executing the Batch Script for Data Load Rules	6-19
Setting the Parameters for Data Load Rules	6-20
Executing the Batch Script for Metadata Rules	6-21
Setting the Parameters for Metadata Rules	6-21
Executing the Batch Script for HR Data Rules	6-21
Setting the Parameters for HR Data Rules	6-22
Executing the Batch Script to Import Mapping Rules	6-22
Setting the Parameters to Import Mappings Rules	6-22
Executing the Batch Script for Data Load Rules to Write Back	6-23
Running a Batch	6-23
Setting the Parameters for Running the Batch	6-24

7 Creating and Using Scripts

Overview	7-1
Key Jython Concepts	7-1
Using the Script Editor	7-2
Overview	7-2
Launching the Script Editor	7-2
Script Editor Layout	7-2
Using Import Scripts	7-2
Overview	7-3
Creating Import Scripts	7-3
Import Script Parameters	7-4
Assigning Import Scripts to Import Formats	7-4
Using Import Script Functions	7-6

Extracting Beginning Characters	7-6
Extracting Middle Characters	7-7
Extracting End of String	7-7
Using Split Functions	7-8
Using the Skip Function (Conditional Skip)	7-9
Storing and Retrieving Temporary Variables	7-9
Storing Temporary Variables	7-9
Retrieving Temporary Variables	7-11
Import Script Sample	7-11
Using Mapping Scripts	7-13
Overview	7-13
Creating Mapping Scripts	7-13
Using Jython Objects with Mapping Scripts	7-15
Mapping Script Samples	7-16
Using Event Scripts	7-17
Overview	7-17
FDMEE Supported Event Scripts	7-17
Creating Event Scripts	7-20
Stopping the Execution of FDMEE from within a Script	7-21
Dynamically Changing Import Formats	7-21
Using the File System Object in Event Scripts	7-21
Events Script Sample	7-22
Using Custom Scripts	7-25
Overview	7-25
Creating a Custom Script	7-25
Working with Custom Scripts	7-25
Adding a Custom Script Group	7-26
Registering Scripts	7-26
Executing a Custom Script	7-28
Executing a Custom Script from a Command Line	7-28
Custom Script Sample using Jython	7-28
Submitting a Data Load Rule	7-28
Submitting a Report	7-30
Using the JAVA API	7-31
Overview	7-31
JAVA API List	7-33
Working with UNICODE Characters in Jython Scripts	7-42
Using JAVA IDE to Develop Scripts	7-43
Using Visual Basic API and Samples	7-44
Overview	7-44
Registering the Visual Basic API DLL Manually	7-44

Visual Basic API List	7-45
API Class (clsAppServerDM)	7-45
Application Class (clsApp)	7-45
Connection Class (clsConnection)	7-46
DataManipulation Class	7-47
Utilities Class (clsUtilities.cls)	7-48
Data Access Class (clsDataAccess)	7-48
SQL Manager Class (clsSQLMgr)	7-54
SQL Class (clsSQL)	7-58
DB Tools Class (clsDBTools)	7-59
Maintenance Manager Class (clsMaintenanceMgr)	7-61
POV Manager Class (clsPOVMgr)	7-62
State Class (clsState)	7-64
Visual Basic Sample Scripts	7-66
Visual Basic Script to Send Email	7-66
Visual Basic Script to Copy a File to the Inbox	7-69
Visual Basic Script to Execute SQL	7-71
Visual Basic Script to Access Various Properties	7-73

8 FDMEE Reports

FDMEE Reports	8-1
Working with Query Definitions	8-2
Working with Report Definitions	8-3
Adding Report Groups	8-3
Working with Report Definitions	8-4
Creating a Report Template	8-6
Creating a XLIFF File for Translated Report Templates	8-7
Running Reports	8-8
FDMEE Detail Reports	8-10
Audit Reports	8-10
Account Chase Wildcard (TargAcct, Per, Cat)	8-10
Account Chase - Freeform (TargAcct, Per, Cat)	8-11
Map Monitor for Location	8-11
Map Monitor for User	8-12
Check Reports	8-13
Check Report	8-13
Check Report Period Range (Cat, Start Per, End Per)	8-14
Check Report With Warnings	8-14
Check Report By Validation Entity Seq.	8-14
Base Trial Balance Reports	8-15

TB Current Location, with Targets (Cat, Per)	8-15
TB Current Location with Rules (Cat, Per)	8-15
TB Current Locations, All Dimensions-Targets, by Target Entity-Account (Cat, Per)	8-16
TB Current Locations, All Dimensions-Targets (Cat, Per)	8-16
TB Current Location, by Target Acct (Cat, Per)	8-17
TB Current Location, By Target Entity Account (Cat, Per)	8-17
TB Converted Current Location by Target Entity/Account	8-17
Listing Reports	8-18
Import Formats by Location	8-18
Location Listing	8-18
Location Analysis	8-19
Dimension Map (Dimension)	8-19
Dimension Map For POV (Dimension, Cat, Per)	8-19
Process Monitor Reports	8-19
Process Monitor (Cat, Per)	8-20
Process Status Period Range (Cat, Start Per, End Per)	8-20
Process Monitor All Categories (Cat, Per)	8-20
Variance Reports	8-21
Account Chase Variance	8-21
Trial Balance Variance	8-21

A FDMEE Web Service

executeDataRule	A-1
executeMetaDataRule	A-2
getDataRuleNames	A-2
getLocationNames	A-3
getPeriodNames	A-3
lockPOV	A-3
unlockPOV	A-4

B FDMEE REST APIs

URL Structure for FDMEE	B-1
Running Data Rules	B-1
Running Batch Rules	B-4
Import Data Mapping	B-6
Export Data Mapping	B-8
Execute Reports	B-10

C	Source System Tables Used by FDMEE	
	Fusion and E-Business Suite Source System Tables	C-1
	PeopleSoft Enterprise Financial Management Source System Tables	C-3
	PeopleSoft Human Capital Management Source System Tables	C-5
	PeopleSoft Commitment Control Source System Tables	C-5
	Creating Synonyms for E-Business Suite Tables	C-6
D	Creating an Alternate Schema in an Enterprise Resource Planning (ERP) Source System	
E	Staging Tables	
	FDMEE Staging Tables	E-1
	Staging Table Used for Import from Source	E-2
	Data Tables Used for Drill Through	E-2
	Views Used for Export to Target	E-2
	PeopleSoft Human Capital Management Staging Tables	E-3
	PS90HCM	E-3
	CHARTFIELD	E-15
F	Archiving the TDATAMAPSEG Table	
G	Working with Open Interface Adapters	
	Setting up the Open Interface Adapter	G-1
	Using the Open Interface Table	G-1
	Loading Source Data using Custom Loaders	G-2
	Customizing ODI to Load Data from a Custom Source	G-3
	Migrating the Customization	G-4
H	PeopleSoft Commitment Control	
I	Report Query Table Reference	
	TDATASEG Table Reference	I-1
	TLOGPROCESS Table Reference	I-5

J System Maintenance Tasks

Maintain Application Folder	J-1
Maintain Process Table	J-2
Maintain EBS GL Balances Table	J-2
Maintain PeopleSoft Ledger Table	J-2
Maintain FDMEE Data Tables	J-3
Executing Purge Scripts	J-3

K Setting up Jython, Eclipse and Python

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

Send feedback on this documentation to: epmdoc_ww@oracle.com

Follow EPM Information Development on these social media sites:

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

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

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

Google+ - [https://plus.google.com/106915048672979407731/
#106915048672979407731/posts](https://plus.google.com/106915048672979407731/#106915048672979407731/posts)

YouTube - <https://www.youtube.com/user/EvolvingBI>

1

Using FDMEE

Oracle Hyperion Financial Data Quality Management, Enterprise Edition enables you to perform the following tasks:

- integrate metadata and data from an Enterprise Resource Planning (ERP) source system, data from a file, or an Open Interface table into an Enterprise Performance Management (EPM) target application.
- integrate Oracle General Ledger data with an EPM application if you use Fusion Cloud Release 11 or higher.

FDMEE also supports the Financials Accounting Hub (FAN) and the Financial Accounting Hub Reporting Cloud Service (FRACAS) as part of its integration with the Oracle General Ledger.

- drill through from the EPM target application and view data in the Enterprise Resource Planning (ERP) source system.
- synchronize data between EPM applications—Move data between EPM applications, for example, copying data from Financial Management to Essbase for reporting.
- write back budget data to the source system from any Oracle Hyperion Planning, Oracle Essbase aggregate storage, or Essbase block storage application.
- write back data from Oracle Hyperion Financial Management to Oracle E-Business Suite.

For information on supported EPM System versions, see the *Oracle Enterprise Performance Management System 11th Release Certification Matrix*.

For FDMEE issues and workaround, see the *Oracle Hyperion Financial Data Quality Management Readme*.

Supported Source Systems

Oracle Hyperion Financial Data Quality Management, Enterprise Edition supports general ledger data from:

- File-based data loads that enable users to import balances from delimited or fixed-width text files.
- Text files and Excel files
- Oracle E-Business Suite 11i
- Oracle E-Business Suite 12
- Oracle Financials Cloud
- PeopleSoft Enterprise Financial Management 9
- PeopleSoft Commitment Control
- SAP ERP Financial

- SAP BW (Business Warehouse)
- JD Edwards General Ledger system

FDME supports human resource data load for PeopleSoft Human Capital Management 9.

Integration includes data load and drill through. The integration is supported for the following SAP ERP Financial modules as well:

- Classic and new General Ledger
- Profit Center
- Cost Center
- Vendor Balances
- Customer Balances

In addition FDME provides support for Open Interface Adapter. The Open Interface Adapter enables you to import data from any source system using the interface table.

For information on supported technologies for each source system, see the *Oracle Hyperion Enterprise Performance Management System Certification Matrix*.

File-Based Data Loads

File-based imports and write back are supported for those users who do not have a direct connection to their Enterprise Resource Planning (ERP) source data, but have data available from their sources in a text file. Any file, whether it is a fixed width file or a delimited file, can be easily imported into the target EPM application. For example, you can take a trial balance report generated from your source system, and map it into Oracle Hyperion Financial Data Quality Management, Enterprise Edition by way of the import format feature. You can instruct the system where the account, entity, data values and so on reside in the file, as well as which rows to skip during the data import. This feature enables a business user to easily import data from any source, and requires limited technical help, if any, when loading into a target application.

You can also define data load rules which determine how you want to extract or write back data from an EPM system to a file system. For example, you may want to write back budget data.

Supported EPM System Target Applications

Supported Oracle Enterprise Performance Management System target applications (Oracle Hyperion EPM Architect and Classic) are:

- Oracle Hyperion Planning
- Oracle Hyperion Public Sector Planning and Budgeting
- Oracle Hyperion Financial Management (including Tax Provision Application)
- Oracle Essbase aggregate storage and Essbase block storage—If Essbase has been deployed in standalone mode, it must be registered with Shared Services because it does not work directly with Oracle Hyperion Financial Data Quality Management, Enterprise Edition. Planning applications that rely on the standalone Essbase application cannot work directly with FDME either.
- Oracle Hyperion Profitability and Cost Management

- Account Reconciliation Manager (ARM)

Key FDMEE Benefits

Key benefits supported in Oracle Hyperion Financial Data Quality Management, Enterprise Edition include:

- Improved User Experience—Integrated in Oracle Enterprise Performance Management System. The FDMEE user interface is consistent with the Oracle Hyperion Planning and Oracle Hyperion Financial Management user interfaces.
- Improved performance—Improved user interface and data load performance
- Close Oracle Hyperion Shared Services integration—All native Shared Services features are supported (for example, user groups).
- Support for Internet Explorer and Firefox browsers
- Support for Oracle Hyperion Enterprise Performance Management System Lifecycle Management—Consistent Lifecycle Management support like other EPM System products

You can migrate a Lifecycle Management application from release 11.1.2.3 to release 11.1.2.4.

- Support for multiple platforms—All supported platforms.
- Consistent Scaling and Load Balancing—All scale out configuration and load balancing steps are consistent with other EPM System products.

Also see the *Oracle® Enterprise Performance Management System Deployment Options Guide*.

- FDMEE can be used as a primary gateway to integrate on-premise and cloud based applications. This

as a primary gateway to integrate on-premise and cloud based applications. This feature EPM customers to adapt cloud deployments into their existing EPM portfolio.

- Data synchronization that enables you to easily move data between the EPM applications irrespective of the dimensionality of the application without having to create a data file from the EPM source application. You specify the source and target EPM application, and then map the data. Given the powerful mapping features already available, the data can be easily transformed from one application to another application. For example, synchronizing data enables you to move data from Financial Management to Oracle Essbase for reporting.
- Write-back support from all EPM applications (except the Accounts Reconciliation Manager) to Enterprise Resource Planning (ERP) applications. This feature offers significant advantages, such as writing back budgets created in Planning to Peoplesoft or E-Business Suite ERP GL, or moving adjustment journals from Financial Management to ERP Systems like Oracle E-Business Suite or Peoplesoft. Other target systems still need to use a custom application framework.

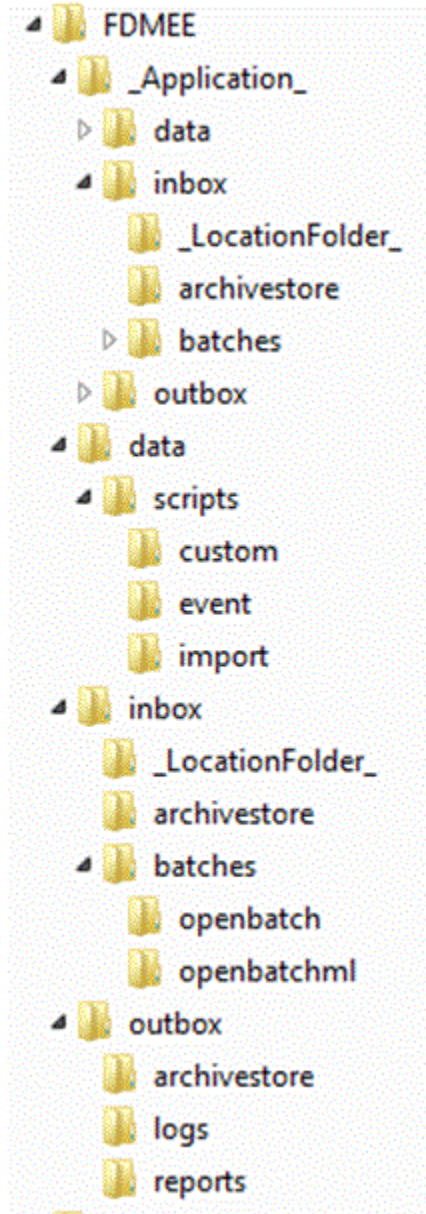
FDMEE Application Folder Architecture

Oracle Hyperion Financial Data Quality Management, Enterprise Edition uses a set of folders that are created under the application root folder. The application root folder is

created by the system administrator on the server where FDMEE is installed, and then defined in the System Settings under Configure in FDMEE.

To create the basic file structure click the **Create Application Folder** button from the System Settings screen. In addition application folders and location folders can also be created. If application folders are not used, then location folders are created in the top level inbox. When application folders are used, then location folders are created in the related application folder.

Here is an example of a possible folder structure:



The standard FDMEE application uses the following folder structure:

Table 1-1 FDMEE Application Folder Structure

Folder	Descriptions
data	The data folder contains a copy of each file that is loaded by the system, and is also the root folder for the scripts directory. Each file in this folder is assigned a unique name, and can be opened from the Data Load Workbench by clicking an amount or from the Drill-through screen.
scripts	This is the top folder for the custom, event, and import folders. Scripts of these types are stored in these folders.
custom	custom—Contains custom scripts written in Visual Basic or Jython script.
event	event—Contains scripts that are executed for the specified system event and are written in Visual Basic or Jython script.
import	import—Contains scripts that are associated with an import format and are executed during the import processing step. These scripts are only written in Jython.
inbox	You can use the inbox default directory from which to import source files, as a central repository for all ledger extract files. Because source files can be retrieved from any accessible directory, you are not required to place import files in this directory. When you select the option to create a folder for every location created by the user, they are created in the inbox. The inbox includes the batches and archivestore directory.
archivestore	Reserved for future use.
batches	batches is a top level folder only and not used to store files.
openbatches	openbatches is the location where files are stored by the system. These files contain the "Point of View" in the name so that one or more files can be picked up by the system and loaded without needing to set the POV in the user interface. This is very useful when a large number of files need to be loaded at a single time.
openbatchesm1	Similar to the openbatches folder; however, these files contain multiple periods in a single file.
outbox	The outbox folder stores export files created by FDMEE, any error log files from a target application, drill region load files, and drill load logs from the target.

Table 1-1 (Cont.) FDME Application Folder Structure

Folder	Descriptions
archivestore	Reserved for future use.
logs	Stores logs generated by the load processes in the format EPM-APPLICATION-NAME_PROCESS-ID .log. These logs can be viewed using the Show Log link in the Process Details page of FDME.
reports	The reports folder stores report output generated by the batch reporting process in PDF, HTML or XLS. Also includes output from reports run in offline mode.

**Note:**

To create the folder structure for applications, define the root folder in the Application Settings screen, and then select the **Create Application folders** option. For example, when you want to run a set of scripts for a specified application, then the use of the application folders is required.

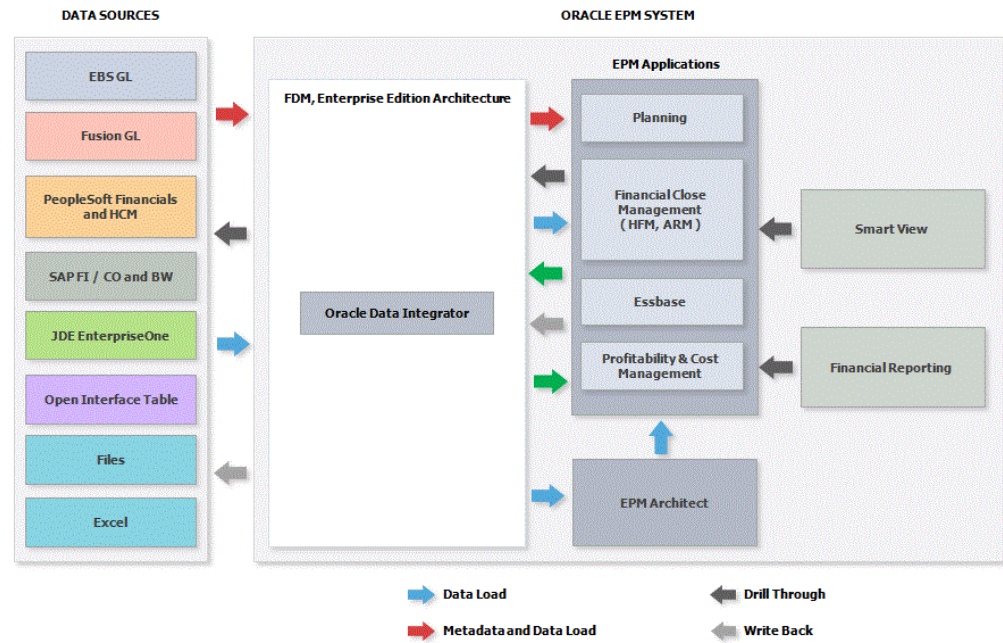
FDME Architecture

Oracle Hyperion Financial Data Quality Management, Enterprise Edition is the key application for integrating Enterprise Resource Planning (ERP) systems with Oracle's Hyperion EPM applications. FDME is accessed through Oracle Hyperion Enterprise Performance Management Workspace, which uses Oracle Hyperion Shared Services to authenticate users. The key to its integration lies within its underlying engine, Oracle Data Integrator.

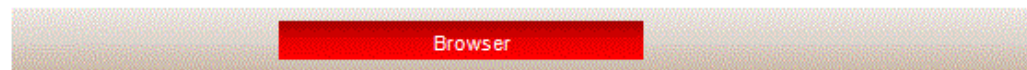
FDME sits on top of Oracle Data Integrator and orchestrates the movement of metadata and data into EPM applications. The application server can be deployed on multiple platforms (see the *Oracle Hyperion Enterprise Performance Management System Certification Matrix*) and connects with EPM applications such as Oracle Hyperion Financial Management, Oracle Hyperion Planning, Profitability, Account Reconciliation Manager, and Oracle Essbase.

The ODI Agent installed and used by FDME is exclusively for processes launched as part of FDME processing, including the FDME user interface, FDME batches, or FDME jobs executed by way of a Windows or Lynx batch. The FDME repository is for exclusive use of FDME objects shipped by Oracle, or for castigations made by the customer to these objects. The ODI agent or repository used by FDME cannot be used for any other purpose by the customer. Any other ODI work needs to use a separate agent and repository.

The following diagram shows the technical architecture structure of FDME:



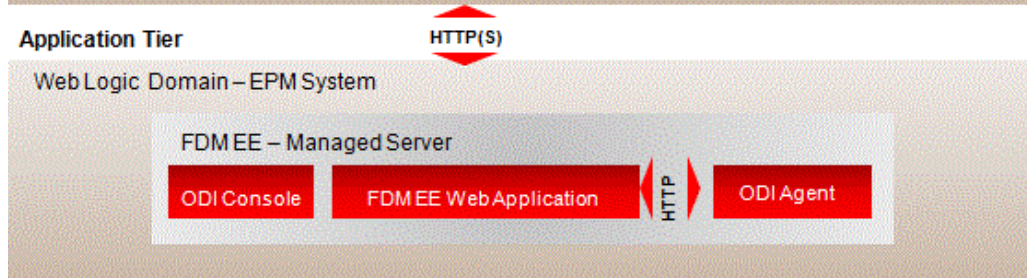
Client



Web Tier



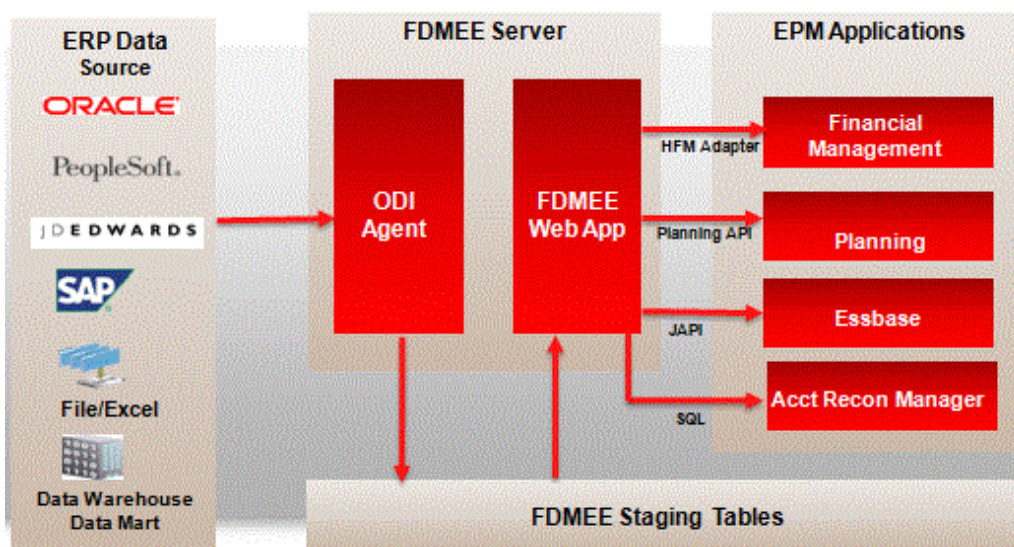
Application Tier



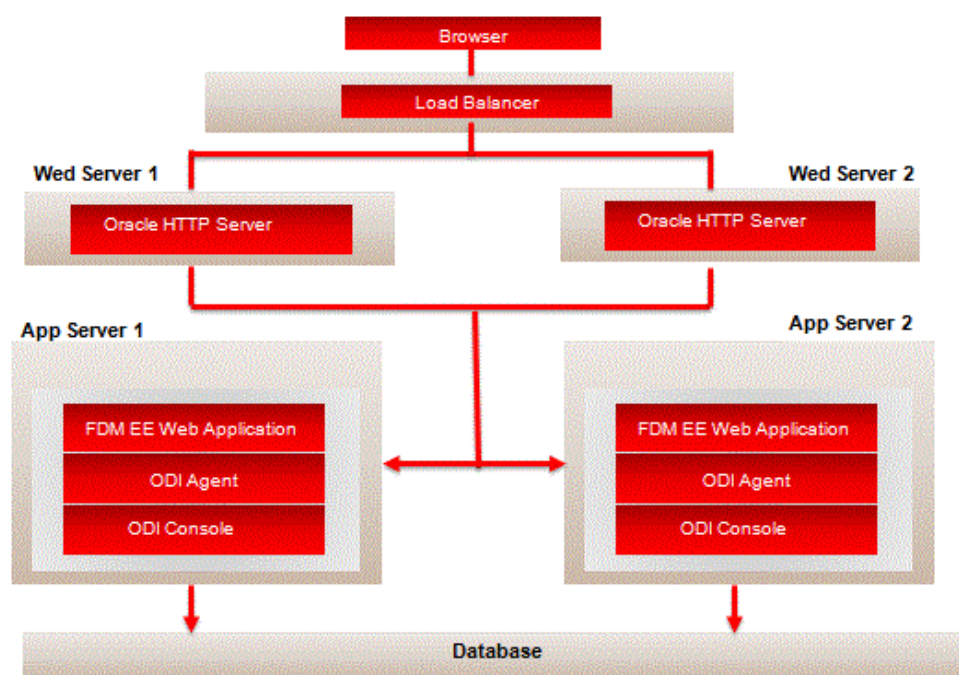
Database Tier



The following diagram shows the flow of data in FDMEE:



The following diagram shows FDMEE High Availability:



Drilling Through

Oracle Hyperion Financial Data Quality Management, Enterprise Edition provides the framework to drill through from EPM applications back to the general ledger source. Drill through is not supported for human resource data. Users can drill through to detail in the source system through FDMEE.

The ability to create a drill region is enabled in Target Application options. FDMEE creates drill region by scenarios. For any cube (Planning plan types or Essbase

databases, the name of the drill region is `FMEE_<name of the scenario member>`. When creating the drill region, FMEE checks if a dimension is enabled for the drill. Members of enabled dimensions selected in data loads, are included in the drill region filter. If no dimensions are enabled, the following dimensions are enabled by default: Scenario, Version, Year, and Period. You can enable additional dimensions, and the subsequent data load considers members of newly enabled dimensions. If you disable any dimensions which were previously included in a drill region used for drill creation, members of such dimensions are not deleted during the subsequent data loads. If needed, you can remove obsolete members manually.

 Note:

If the source system is Oracle E-Business Suite/PeopleSoft and you have metadata rules, then the drill region is created based on the metadata rule. Otherwise, it is created based on the target members in the data load mappings. For Year, Period, and Scenario, FMEE uses audit information to create the drill region.

 Note:

In Oracle Smart View for Office and Oracle Hyperion Financial Reporting, you can drill through only if the data source is Oracle Hyperion Financial Management, Oracle Hyperion Planning, Oracle Essbase, and Oracle Hyperion Profitability and Cost Management.

 Note:

Drill through is not supported for Financial Management journals and intercompany transactions.

When you drill through, if data was loaded by FMEE, a landing page is displayed in a new Oracle Hyperion Enterprise Performance Management Workspace tab or a new window. The landing page is a gateway to the data in the source system. See [Drilling Through to the FMEE Landing Page](#).

Drilling Through to the FMEE Landing Page

The Oracle Hyperion Financial Data Quality Management, Enterprise Edition landing page displays general ledger accounts and the hyperlinked balances that were used to populate the cells in the EPM application. When you click a linked data value, you can drill through to the source system and view the associated journal entries for the selected general ledger account.

You can drill through to balances to display data loaded from your source system. When you navigate to the Oracle General Ledger Balances page after login validation, you can view a table listing the general ledger accounts that contributed to the drilled value shown in the EPM application for the specific period.

This table includes a breakdown of all general ledger accounts values with hyperlinks, enabling users to further drill into the Journal Lines page in Oracle General Ledger. Users can then view the associated journal entries for the selected Oracle General Ledger account.

When you navigate to PeopleSoft Enterprise Financial Management, the Ledger Inquiry page is displayed after login validation. Users can then view information on the Journal Inquiry page. See *PeopleSoft Enterprise General Ledger 9.1 PeopleBook* for additional information on drill through capabilities.

Configuring Oracle Data Integrator With FDMEE

Oracle Hyperion Financial Data Quality Management, Enterprise Edition relies on Oracle Data Integrator as the engine that extracts data and metadata from the defined sources, and then populates the Performance Management Architect interface tables or populates the Classic applications with the extracted artifacts. The Oracle Hyperion Enterprise Performance Management System Installer installs ODI when FDMEE is installed. The installer also configures the ODI Master and Work Repository and ODI J2EE Agent as part of install process.

In addition, you must perform some manual configuration steps in Oracle Data Integrator before using FDMEE.



Note:

You only need to configure ODI when you load data from a source other than a file. File-based data loads work out of the box without additional ODI configurations.

1. Set up the data server based on the Enterprise Resource Planning (ERP) source system.
2. Set up the Physical Schemas.
3. Set up the ODI Context Code.

Set up the Data Server Based on the Enterprise Resource Planning (ERP) Source System

You must set up the appropriate data servers based on the Enterprise Resource Planning (ERP) source system used to source metadata and/or data.

When importing from Enterprise Resource Planning (ERP) sources (Oracle E-Business Suite/PeopleSoft/Fusion/JDE/SAP), set up the applicable data servers listed below:

- EBS_DATA_SERVER—For E-Business Suite General Ledger
- PSFT_FMS_DATA_SERVER—For PeopleSoft General Ledger and Commitment Control
- PSFT_HCM_DATA_SERVER—For PeopleSoft Human Capital Management (HCM)

- FUSION_DATA_SERVER—For Fusion General Ledger
- JDE_DATA_SERVER—JD Edwards Enterprise (JDE) General Ledger
- SAP_SERVER—SAP FICO

To update server connections information:

1. Access the Oracle Data Integrator Console.
2. Select the **Browse** tab.
3. Expand **Topology**.
4. Select the operating data server to update, and then click **Edit**.
For example, select **EBS_DATA_SERVER** or **PSFT_FMS_DATA_SERVER**.
5. In **Edit Data Servers**, under **JDBC Details**, enter the JDBC driver in **JDBC Driver**.
For example, enter `oracle.jdbc.OracleDriver`
6. In **JDBC URL**, enter the JDBC URL address.
For example, enter `jdbc:oracle:thin:@<host>:<port>:<sid>`
7. In **User**, enter the user name.
8. In **JDBC Password**, enter the password.
9. Click **Save**.

Setting up Physical Schemas

To update a physical schema:

1. Access the Oracle Data Integrator Console.
2. Select the **Browse** tab.
3. Expand **Schemas**.
4. Expand **Physical Schemas**.
5. Select the schema to update, and then click **Edit**.
For example, select **EBS_DATA_SERVER** or **PSFT_FMS_DATA_SERVER**.
6. In **Schema Name**, enter the schema name in uppercase characters.
7. Click **Save**.

 Caution:

Extensive problems can occur if you switch the Enterprise Resource Planning (ERP) system connection information for the Physical Schema in the Oracle Data Integrator Topology Manager after you have completed initial configuration. For example, extensive problems can occur if you start using one physical schema (ERPTEST) pointing to ERP Test Instance1 in the Oracle Data Integrator Topology Manager, and then change to a connection information in this physical schema to point to ERP Test Instance2 without first creating a new context in Oracle Data Integrator. The correct procedure is to create two physical schemas (ERPTEST1 and ERPTEST2) each pointing to a different ERP instance. Then, create two contexts and associate the appropriate physical schema to the logical schema in the context.

Setting up the ODI Context Code

You can set up the default "GLOBAL" ODI context code. The ODI context code refers to the context defined in Oracle Data Integrator. A context groups the source and target connection information.

To set up the default "GLOBAL" ODI context code:

1. Access the Oracle Data Integrator Console.
2. Select the **Browse** tab.
3. Expand **Contexts**.
4. Select **Global** update, and then click **Edit**.
5. In **Edit Context Global**, in **Context Name**, enter GLOBAL.

Enter the word "GLOBAL" in uppercase characters only, that is, enter **GLOBAL**.

6. Select **Default Context**.
7. Click **Save**.

Setting up ODI to Integrate with JD Edwards

To set up ODI to integrate with JD Edwards:

1. Download the **JD Edwards EnterpriseOne Data Access Driver (DAD)** driver:
 - a. Display **My Support** by clicking [My Oracle Support](#).
 - b. Select the **Patches and Updates** tab, and then select **JD Edwards Patches**.
 - c. From **JDEdwards**, enter EnterpriseOne Tools Releases.
 - d. In **Release**, select **All Releases**.
 - e. In **Platform**, select **Multiplatform**.
 - f. In **Description** (*text*), enter ***Data Access Driver*** .
Include asterisks as it is a wildcard search.
 - g. Select the **License Agreement** check box.
 - h. Click **Search**.

- i. Click the plus sign (+) to add **Tools 9.1,* Data Access Driver** to the Download Basket.



Note:

Oracle Hyperion Financial Data Quality Management, Enterprise Edition supports JD Edwards Tools 9.1 and 9.2. If you use JDE Tools 9.2.x.x, then use the 9.2.x.x DAD Driver. If you use JDE Tools 9.1.x.x, then use the 9.1.x.x DAD Driver.

- j. Click the Item(s) hyperlink and download the tools release specific item.
2. To copy the DAD driver:

The DAD driver has a .par extension, but it can be unzipped.

 - a. Unzip the archive file to a temporary directory.
 - b. Extract (or unzip) the JAR file DADriver_EAR.jar.
 - c. Copy the extracted contents of the DADriver_EAR.jar to the EPM_MIDDLEWARE_HOME\odi\odi_misc directory.

Typically, EPM_MIDDLEWARE_HOME is C:\Oracle\Middleware
 3. Request that the JD Edwards system administrator provide you with the following files:
 - jdbj.ini
 - jas.ini
 - jdelog.properties

These files are generated when you register the JD Edwards EnterpriseOne Data Access Driver using the JD Edwards Server Manager.
 4. Copy the jas.ini, jdbj.ini, and jdelog.properties files from the **JDE Application Server** to the EPM_MIDDLEWARE_HOME\odi\odi_misc directory.
- Typically, EPM_MIDDLEWARE_HOME is C:\Oracle\Middleware
5. Ensure that the **security server** of the jas.ini file is set up correctly.
 6. Review and edit the jdbj.ini file as follows:
 - a. If the JD Edwards application database is Oracle, update the location in the tnsnames.ora file:

For example, modify tns= C:\Oracle\Middleware\user_projects\config\dbclient\tnsnames.ora in the jdbj.ini file.

Forward or backward slashes can be used.
 - b. Encrypt the {JDBJ-BOOTSTRAP SESSION} stanza, and then the password value.

If the stanza has not changed, no update is required.

 - c. Modifications to the .INI file password encryption can be accomplished using the Server Manager.
 - d. Set the value of the following parameters under [JDBJ-RUNTIME PROPERTIES] stanza to:


```
resultSetTimeout=-1
```

```
transactionTimeout=-1

usageExecutionThreshold=20000

usageResultSetOpenThreshold=120000

usageTracking=false

msSQLQueryTimeout=1800000
```

e. Edit the tnsnames.ora file.

If the tnsnames.ora entry exists, copy and rename it.

The tnsnames.ora file must include the reference to the JDE server, for example,

```
jra91dl =

  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = dene1lx7.us.oracle.com)(PORT = 1530))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = jra91dl)
    )
  )
```

The tnsnames.ora file is located in C:\Oracle\Middleware\user_projects\config\dbclient\tnsnames.ora.

7. Edit the jdelog.properties file, and set the log level as needed in all references in the file.

In a production environment, set the log level to SEVERE so that only severe errors are reported in the log file.

8. Change the file path in jdelog.properties for log to C:\Oracle\Middleware\user_projects\epmsystem1\tmp

For example, specify: FILE=C:\Oracle\Middleware\user_projects\epmsystem1\tmp\at9062056.log in all instances found in the file.

9. Ensure that you can access the server specified in the jdbj.ini file from the ODI Server.

You can do so from the command line by issuing a ping to the server.

10. Make sure that environmental variables JAVA_HOME and TEMP are set up correctly.

JAVA_HOME is an environmental variable and points to java.exe under C:\Oracle\Middleware\jdk160_35\bin

TEMP is a user variable and points to %USERPROFILE%\AppData\Local\Temp.

Setting environmental variables might require restarting the machine.

11. If you use JDE Tools 9.1.2 or 9.1.3, run the script copyfilesJDE912.bat / sh from the following directory:

EPM_ORACLE_HOME \products\FinancialDataQuality\bin

If you use JDE Tools 9.1.4 or later, run the script: `copyfilesJDE914.bat / sh` from the `EPM_ORACLE_HOME \products\FinancialDataQuality\bin` directory.

When you run the script, the following is displayed:

```
C:\Users\hitinstall>C:\Oracle\Middleware\EPMSys11R1\products\FinancialDataQuality\bin\copyfilesJDE914.bat
```

```
"Copy required JAR Files"
```

```
1 file(s) copied.
```

```
1 file(s) copied.
```

```
"Script completed successfully"
```

12. If JD Edwards uses Microsoft SQL Server or IBM databases, then download the **JDBC** driver and copy it to the *DOMAIN HOME/lib* directory.

Typically this is the `C:\Oracle\Middleware\user_projects\domains\EPMSys11R1\lib` directory.

Refer to the JD Edwards Tools documentation for the supported versions of the JDBC driver for the database platform.

13. Look for the following information in `jdbj.ini` and verify that it is in the `tnsnames.ora` file.

```
databaseType=0
```

```
serverPort=1530
```

```
name=System - 910
```

```
database=jra91dl
```

```
server=dene1lx7
```

```
physicalDatabase=
```

```
owner=SY910
```

```
lob=true
```

```
unicode=true
```

Specifically, the database name is referenced in the `tnsnames.ora` file.

14. In Oracle Data Integrator, perform a JDE Test Connection with the ODI Agent.

- a. From the **ODI Studio**, launch the **Oracle Data Integrator**.

- b. From **Physical Architecture**, then **Technologies**, and then **Oracle**, select **JDE_Data_Server**.

Test the JDE Test Connection with the ODI Agent and not the local Agent.

- c. Click **Test Connection**.

- d. From **Test Connection for**, select **OracleDI Agent**, and then click **Test**.

- e. From **Physical Architecture**, then **Agents**, then **OracleDI Agent**, right click, and select **Test**.

An Information window displays when the test is successful.

- f. Search for "environment" in the `jdbj.ini` file and note the environment values:

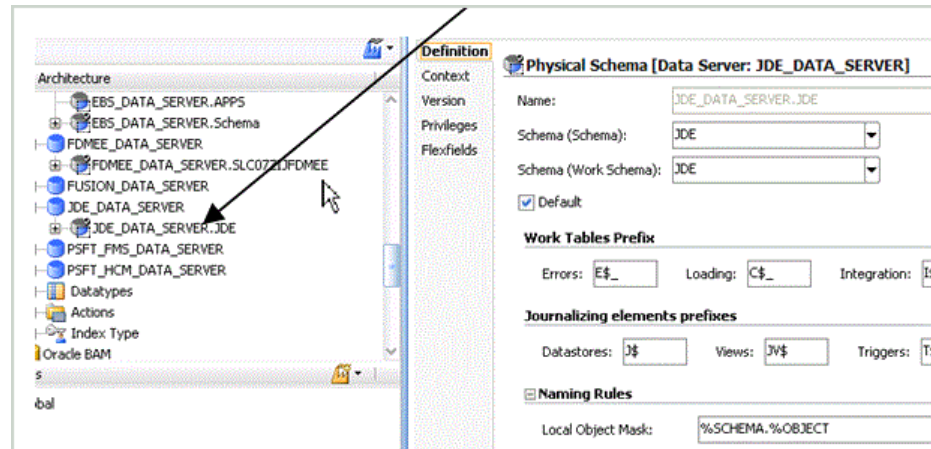
```
[JDBj-BOOTSTRAP SESSION]

user=EOSVM

password=ACHCJKEBHCJKBKEGLDDFKBCLBCDBCGBDCMJMBPGKLN0

role=*ALL

environment=PY910
```
- g. From **ODI Studio**, then **Topology**, then **Technologies**, and then **Oracle**, select **JDE_DATA_SERVER**.



- h. In the **Definition** section, specify the following values:
 - i. In **User**, specify **JDE_USER**.
 - ii. In **Password**, specify **JDE_USER**.
 - i. In the **JDBC** section, specify the following values:
 - i. In **JDBC Driver**, specify **JDBC Driver com.jdedwards.jdbc.driver.JDBCdriver**.
 - ii. In **JDBC Password**, specify **JDBC URL jdbc:oracle:enterpriseone://PY910;RMNEQN=1;enterpriseone.role=*ALL**.
 - **JDBC Driver com.jdedwards.jdbc.driver.JDBCdriver**
 - **JDBC URL jdbc:oracle:enterpriseone://PY910;RMNEQN=1;enterpriseone.role=*ALL**
 - j. From **Physical Schema**, select **JDE** for the user.
 Select the second (indented) JDE server which is the Physical Schema.
15. Restart the FDMEE service for the changes to take effect.
 16. Verify the import of the JD Edwards source system.
 If the import did not occur, see [Importing a Source Adapter](#).

Security

Oracle Hyperion Financial Data Quality Management, Enterprise Edition supports the following roles:

Table 1-2 Security Roles and Task Descriptions

FDMEE Roles	Tasks per Role
Administrator	Manages applications and performs any action in FDMEE. When you log in with the Administrator role, all links are visible in the Tasks pane.
Create Integration	Creates FDMEE metadata and data rules. If you have the Create Integration role, you can: • Create, delete, and edit metadata rules and data rules • View process details • Perform period mappings
Drill Through	When you log in with the Create Integration role, these links are visible in the Tasks pane: Data Load, Member Mapping, HR Data Load, Metadata, and Process Detail. Controls the ability to drill through to the source system. In FDMEE, this role controls whether you can drill to the FDMEE landing page, which controls drilling to the source system.



Note:

You *cannot* run rules, view, create, edit, or delete source system registrations, target system registrations, or source accounting entities.

Table 1-2 (Cont.) Security Roles and Task Descriptions

FDMEE Roles	Tasks per Role
Run Integration	Runs existing FDMEE metadata and data rules. If you log in with the Run Integration role, you can: • Run metadata rules or data rules • View process details
HR Integration	Runs Human Resource data rules and fills out runtime parameters. Can view transaction logs.
Intermediate 2-9	Roles 2-9 for intermediate levels are defined by the administrator.

 Note:

You *cannot* view, create, edit, or delete source system registrations, target system registrations or source accounting entities.

FDMEE users who need to extract data from Oracle or PeopleSoft Enterprise Financial Management must be granted this role to run data rules.

When you log in with the Run Integration role, links are visible in the Workflow tasks: Data Load Workbench, Data Load, Member Mapping, HR Data Load, Metadata, and Process Detail.

 Note:

FDMEE users are enabled to define mappings only to target applications to which they have access.

 Note:

Planning and Essbase do not use member-level security at this time.

Integration Process Overview

You can use Oracle Hyperion Financial Data Quality Management, Enterprise Edition to integrate metadata and data from your Enterprise Resource Planning (ERP) source system in these ways:

- [Extracting General Ledger Data](#)
- [Writing Back Data from EPM Applications](#)
- [Integrating Metadata](#)
- [Extracting Human Resource Data](#)

Extracting General Ledger Data

Follow this process to extract general ledger metadata and data and push into target EPM applications:

1. Register source systems in Oracle Hyperion Financial Data Quality Management, Enterprise Edition by adding details for Oracle Data Integrator and FDMEE, which are specific to the source system.
2. Register target applications for use with FDMEE.
3. Select the Source Accounting Entities.
4. Define import formats and locations.
5. Create metadata rules.
6. Create period mappings for Year and Period dimensions.
7. Create category mappings.
8. Create the member mappings and data load rules.
9. Run the metadata rules to import metadata into Oracle Hyperion Financial Management, Oracle Hyperion Planning applications, and Oracle Hyperion Profitability and Cost Management.

If you use Oracle Hyperion EPM Architect, you also deploy or redeploy the applications.

10. Run data rules to extract data from the source system and push it into the target applications. Data and metadata are staged in the FDMEE staging tables, extracted from the source system, and loaded into the target application.

The data loaded is used for multiple purposes by the respective target applications (Planning, Financial Management, or Oracle Essbase). In addition, you can use the sourced data to drill through from web forms in the applications, or Oracle Smart View for Office and Oracle Hyperion Financial Reporting.

Writing Back Data from EPM Applications

Follow this process to write back data from EPM applications to your general ledger source system:

1. Perform steps 1-5 and step 7–8 in [Extracting General Ledger Data](#).
2. Define write-back mappings for required segments or chartfields.

You can select an EPM application as a source, and an Enterprise Resource Planning (ERP) as a target, and this writes to the journal interface table.

3. Run data load rules to push the data from supported target applications (Oracle Hyperion Planning, Oracle Essbase aggregate storage, Essbase block storage, and Oracle Hyperion Financial Management) into your general ledger source system.

 Note:

You cannot write back data to SAP General Ledger and PeopleSoft Human Capital Management source systems.

 Note:

Write-back from Financial Management is limited only to the Oracle E-Business Suite General Ledger. To write back data from an EPM to legacy or any other unsupported Enterprise Resource Planning (ERP) system, Oracle suggests that you extract data to a custom target application. After extracting the data, convert the data files to a format acceptable to those Enterprise Resource Planning (ERP) systems and import them there.

4. Load the data into E-Business Suite or PeopleSoft Enterprise Financial Management by running a process in Oracle General Ledger or PeopleSoft General Ledger.

Integrating Metadata

Related Topics

- [Understanding General Ledger Integration](#)
- [Requirements](#)
- [Required Dimensions](#)
- [Member Properties Sourced from the Enterprise Resource Planning \(ERP\) System](#)
- [How Dimensions Are Processed](#)
- [How Languages Are Processed](#)
- [How Currencies Are Processed](#)
- [Defining Metadata Rules](#)
- [Loading Source System Hierarchies into EPM Dimensions](#)

Understanding General Ledger Integration

Oracle Hyperion Financial Data Quality Management, Enterprise Edition supports loading data into EPM applications from general ledger source systems and writing back data from target EPM Oracle Hyperion Planning applications to general ledger source systems.

- Loading data from the General Ledger source system—FDMEE supports loading data from general ledger source systems. FDMEE can load both metadata (dimension members and hierarchies from Oracle E-Business Suite, Oracle Fusion Financials and PeopleSoft Enterprise Financial Management only) and data from the Enterprise Resource Planning (ERP) source systems.
- Writing back data to the General Ledger source system—FDMEE enables you to extract data from Planning, Oracle Essbase aggregate storage, Essbase block storage, Oracle Hyperion Financial Management, and then load it into the General ledger source system.

Data load to write-back is unavailable for SAP and JD Edwards. The suggested approach to write back to these Enterprise Resource Planning (ERP) systems and other legacy applications is to extract data to a custom application (data file) and converting them to a format acceptable to ERP systems and importing them as journals.

Requirements

Before you begin using Oracle Hyperion Financial Data Quality Management, Enterprise Edition, consider the following:

- Verify that you have met the EPM *dimension* requirements:
You can build EPM applications with any combination of dimensions. The combination must include required dimensions for the selected application. [Member Properties Sourced from the Enterprise Resource Planning \(ERP\) System](#) describes how member properties are sourced from the Enterprise Resource Planning (ERP) source system.
- Verify that you have met the EPM *member* requirements:
 - Duplicate Members—To avoid issues with duplicate member names, as a best practice, include a unique prefix or suffix for each dimension so each member is always unique.
 - Duplicate Alias Members—If your application has duplicate alias members, it is important to remove any duplicates in the target application or validation errors occur when you deploy the application in Oracle Hyperion EPM Architect.



Note:

Source descriptions must be unique to avoid alias validation errors with Performance Management Architect.

When moving dimensions and members from a source system into a target EPM application, it is important to understand the naming restrictions. For Performance Management Architect, see the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*. For Oracle Hyperion Planning, see the *Oracle Hyperion Planning Administrator's Guide*. For Oracle Hyperion Financial Management, see the *Oracle Hyperion Financial Management Administrator's Guide*.

Required Dimensions

You can build EPM applications with any combination of dimensions, as long as the combination includes those required for the selected application. For example, Oracle Hyperion Planning requires different dimensions to be present in an application than in Oracle Hyperion Financial Management.

For detailed information on dimensions required and properties for Oracle Hyperion EPM Architect applications, see the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*. For Classic Planning applications, see the *Oracle Hyperion Planning Administrator's Guide*. For Classic Financial Management applications, see the *Oracle Hyperion Financial Management Administrator's Guide*. For Classic Oracle Essbase, see the *Oracle Essbase Database Administrator's Guide*.

The following dimensions require special considerations when integrating with Oracle Hyperion Financial Data Quality Management, Enterprise Edition:

- Account
- Currency
- Entity
- Scenario
- Version
- View
- Year
- Period

In addition to the above list, review properties set by FDMEE in the Custom dimension. See [Custom](#).

For information on special considerations and requirements for Oracle Hyperion Public Sector Planning and Budgeting, see [Loading Human Resources Data](#).

Member Properties Sourced from the Enterprise Resource Planning (ERP) System

For each required dimension, specific properties must be defined. The required dimension properties relate to Oracle Hyperion Planning, Oracle Hyperion Financial Management, or Oracle Essbase applications, and in some cases both.



Note:

Oracle Hyperion Financial Data Quality Management, Enterprise Edition sets some of the required properties, but not all.

Account

The Account dimension represents a hierarchy of natural accounts. Accounts store financial data for entities and scenarios in an application. Each account has a type,

such as Revenue or Expense, that defines its accounting behavior. The Account dimension is mapped from the source accounting entity to the EPM Account dimension as defined in the dimension mapping definition for the selected chart of accounts or business unit. The properties set by Oracle Hyperion Financial Data Quality Management, Enterprise Edition are shown below. (Any properties not set are defaulted by the application or Oracle Hyperion EPM Architect).

Table 1-3 Account Dimension Required Properties

Property	Application Type	Population Method/Value
Consolidation Account Type	Consolidation	Populated from the account type in the source accounting entity with the domain of revenue, expense, asset, or liability. If source type is equity, it is changed to liability for use by Oracle Hyperion Financial Management applications.
Account Type	Planning	Populated from the account type in the source accounting entity with the domain of revenue, expense, asset, liability, or equity.
Variance Reporting	Planning, Essbase aggregate storage, and Essbase block storage	Set to "Expense" if account type is expense; otherwise set to "NonExpense." (NonExpense is the default).
Description, Display String	System	Populate from the source accounting entity description.
Time Balance	Planning, Essbase aggregate storage, Essbase block storage	For income statement accounts, (revenue and expenses) set to "Flow" for Planning applications. For Essbase aggregate storage and block storage applications, set to "Last." For balance sheet accounts (asset, liability, and equity), set to "Balance." These properties can be set when you create metadata rules. See Defining Metadata Rules .

Entity and Intercompany

The Entity dimension represents the organizational structure of the company, such as the management and legal reporting structures. Entities can represent divisions, subsidiaries, plants, regions, countries, legal entities, business units, departments, or any organizational unit. You can define unlimited entities.

The Intercompany dimension represents all intercompany balances that exist for an account. This is a reserved dimension that is used in combination with the Account dimension and any custom Oracle Hyperion Financial Management dimension.

Financial Management requires that members of the Entity dimension have the IsICP property set for those members that are intercompany members. When an application is populated, Oracle Hyperion EPM Architect populates the ICP (intercompany) dimension with the appropriate members based on the Entity members that are flagged as ICP (intercompany) entities.

E-Business Suite has two scenarios for mapping source segments to the Entity dimension: 1) an intercompany segment exists in the source chart of accounts, and 2) an intercompany segment does not exist in the source chart of accounts. For PeopleSoft, the business unit is mapped to the entity and the affiliate is mapped to the ICP.

The properties set by Oracle Hyperion Financial Data Quality Management, Enterprise Edition are shown below. (Any properties not set are defaulted by the application or Performance Management Architect.

Table 1-4 Entity and Intercompany Required Properties

Property	Application Type	Population Method/Value
Name	Consolidation, System	Populated from the code/value in the source accounting entity.
Description	System	Populated from the name in the source accounting entity.
ISICP	Consolidation	If the intercompany segment exists in the source, then this flag is set automatically per the rules defined. If the intercompany segment does not exist, then you specify how this property is set. See Entity and Intercompany. For ICP transaction data to load correctly, you must manually set the property ISICP="Y" for those accounts participating in ICP. In Performance Management Architect, you can use the Property Grid to modify the property. If using Financial Management Classic application administration, extract the metadata, update, and then re-import it back. After modifying the property, you can load data correctly for ICP transactions.
Currency	Consolidation, Essbase, Planning	For Financial Management target applications: The entity currency is set based on the default defined in the mapping rule for the Entity dimension. (All members are assigned the same currency.) As the administrator, ensure that the functional currency of the source is consistent with the default Entity currency.

 Note:

These are the only properties that are set as part of the FDMEE integration, all others are defaults when you create new members. If a property was originally set by FDMEE, and you change it later, the property is overridden.

Scenario

The Scenario dimension represents a set of data, such as Budget, Actual, or Forecast. For example, the Actual scenario can contain data from a general ledger, reflecting past and current business operations. The Budget scenario can contain data that reflects the targeted business operations. The Forecast scenario typically contains data that corresponds to predictions for upcoming periods. A Legal scenario can contain data calculated according to legal GAAP format and rules.

Version

The Version dimension is specific to EPM applications and usually does not have a source in the source accounting entity. Since it is required, you must specify the necessary default value in the member mapping by using the "Like" mapping type. When defining the data rule in Oracle Hyperion Financial Data Quality Management, Enterprise Edition, select the desired "Version" to include with the extracted data. Since the Version dimension is not extracted from the source system, it is not necessary to define specific properties.

View

The View dimension represents various modes of calendar intelligence; for example, Periodic, Year-to-Date, and Quarter-to-Date frequencies. Oracle Hyperion Financial Data Quality Management, Enterprise Edition extracts only data that is below the quarter level. You select the view as part of the data rule definition, and when the data is extracted, it includes the View selection as the value for the dimension on each row. See [Defining Data Load Rules to Extract Data](#). Since the View dimension is usually not extracted from the source system, it is not necessary to define specific properties. However, before the data extraction process, you must create all members in the View dimension manually.

Year and Period

The mapping between the source system calendar and the Year and Period dimensions is managed using the period mapping feature described in [Defining Period Mappings](#). Before you perform period mapping, create the necessary Year and Period members. In Oracle Hyperion Financial Data Quality Management, Enterprise Edition, you select the calendar periods to include in the data extraction process and on the Period Mapping page to define the appropriate target year and target period dimensions to assign to the data. Since the Year and Period dimensions are not extracted from the source system, you need not define specific properties.

 Note:

For Oracle Hyperion Planning applications, it is required that you must have the same number of children in each branch of the Period dimension. For example, Q4 has October, November, December children and an adjustment period in Oracle Hyperion EPM Architect.

Alias

For Oracle Hyperion Planning and Oracle Essbase, the Alias dimension or table is required to support languages. Keep in mind these special considerations:

- The Alias dimension must include a member named "Default."
- If the dimension name is not the same as the Alias name in an Oracle Hyperion EPM Architect Planning application, the drill through landing page does not return any data.
- When creating Alias table members in a dimension, define them with the same name that is displayed in Oracle Fusion, E-Business Suite, or PeopleSoft. This is the value of the NLS_LANGUAGE column.

Custom

The properties set by Oracle Hyperion Financial Data Quality Management, Enterprise Edition are shown below. (Any properties not set defaults in the application or in Oracle Hyperion EPM Architect.)

Table 1-5 Custom Dimensions Required Properties

Property	Application Type	Population Method/Value
Name	System	In E-Business Suite, this value is populated from the Segment Name. In PeopleSoft Enterprise Financial Management, this value is populated from the chartfield value.
Description	System	In E-Business Suite, this value is populated from the Segment Value. In PeopleSoft Enterprise Financial Management, this value is populated from the chartfield value.

How Dimensions Are Processed

For Oracle Hyperion EPM Architect applications, the dimension extract process populates the interface tables with dimension members based on the mapping rule details created in Oracle Hyperion Financial Data Quality Management, Enterprise Edition. For Classic Oracle Hyperion Planning and Oracle Hyperion Financial Management applications, the dimension extract process populates dimension members directly into the application based on the mapping rule details created in FDMEE.

As part of the extract process, the dimension members are directly loaded into the target Performance Management Architect dimension, with specific properties defined as described in [Member Properties Sourced from the Enterprise Resource Planning \(ERP\) System](#). In addition to loading dimension members, the related alias entries are also loaded to provide the appropriate language support.

Shared dimensions from EPMA are not selected directly from the shared library for loading. FDMEE presents the dimensions in the target application for inclusion in a metadata rule, and then updates the shared library automatically when updating the

target application. Dimensions in the shared library cannot be directly loaded, but must be included in a target application to enable loading from an Enterprise Resource Planning (ERP) source.

The FDMEE dimension extract process includes:

1. Extracts the general ledger segment or chartfield value sets from the source system.
 - Only general ledger segment value set members or chartfield members that are relevant to the source chart of account segments or chartfields mapped to Classic Financial Management, or Planning are extracted.
 - The members are loaded into a staging table on the target instance. Before loading them into the staging table, FDMEE assigns the segment values a prefix defined for the corresponding EPM application dimension.

2. Processes dimensions mapped to single segments or chartfields.

For Classic applications, dimensions are loaded directly into the target application. The interface tables for dimensions map to a single general ledger segment or chartfield. This consists of filtering the data from the staging table loaded in step 1, on the basis of the segment value set mapped to a particular dimension, and loading the corresponding dimension member interface table and dimension member property array table (for aliases).

In most cases, dimensions are mapped as a single segment in Fusion and E-Business Suite source systems or single chartfield in PeopleSoft source systems from the source chart of accounts to a target dimension and you select the starting node in the source dimension as the basis for the new dimension.

3. Processes the dimensions mapped to multiple segments or chartfields.

For Performance Management Architect, the member interface tables are populated for the dimensions mapped from more than one E-Business Suite general ledger chart of accounts segment or PeopleSoft chartfield. The individual segment values must be concatenated to create the dimension member values.

The dimension extract process creates the required member entries, properties, and alias entries if they exist in the source system, and then applies defaults to those properties if they do not exist. Users should update member properties if the source system value was unavailable or in cases where a different value is desired.

In some cases, you can sometimes create target dimension members based on the concatenation of one or more source segments. When dimensions are mapped as a concatenated segment, the new dimension is created based on a user-defined traversal order of the source hierarchies into the concatenated member target hierarchy.

How Languages Are Processed

As part of the source system registration, Oracle Hyperion Financial Data Quality Management, Enterprise Edition gets the list of available source system languages, in addition to the base language. The base language is typically the language selected when the Enterprise Resource Planning (ERP) source system is installed. Additional available languages that are not the base language are referred to as the “enabled languages.”

Languages from the source system that are mapped to the languages defined in the target application are independent of the languages available for selection via the FDMEE browser selection. The languages available in the browser might be different from the languages available in the Enterprise Resource Planning (ERP) source system and the target EPM application. For information on languages that FDMEE supports, see the *Oracle Hyperion Enterprise Performance Management System Certification Matrix*.

When you register a target application for use with FDMEE, the Default Language column on the Target Application Registration page is used as follows:

- The languages displayed in the Default Language drop-down list are FDMEE supported languages. These languages are mapped behind the scenes to the Enterprise Resource Planning (ERP) source system languages.
- The Alias dimension in Oracle Essbase and Oracle Hyperion Planning applications has a required "Default" member. The FDMEE language that you select when registering a target application is automatically mapped to the "Default" member. Because the FDMEE language is mapped to the source language for the member description, you map the base or enabled source language in the source system to the "Default" alias member. During processing, all other languages are mapped to the other alias members as long as the alias member exactly matches the FDMEE source language for the member description.

 Note:

Language processing is the same for Essbase and Planning applications.

 Note:

Oracle Hyperion Financial Management languages are processed based on the default language that you select on the Target Application Registration page.

See [Registering Target Applications](#).

How Currencies Are Processed

When you define a data rule, you can specify how to extract exchange rates from the Enterprise Resource Planning (ERP) source system. If your target application has the multi-currency option enabled, you can specify how to process exchange rates.

All rates are extracted and inserted into the AIF_HS_EXCHANGE_RATES table. This table is populated using the ISO currency code for each currency from the source system. The ISO numeric code is not used in this processing.

Exchange rates are pushed into Oracle Hyperion Planning or Oracle Hyperion Financial Management based on a match between the ISO currency code in the AIF_HS_EXCHANGE_RATES table and the currencies defined in the multi-currency Planning or Financial Management application. (It is important to set up the currencies in the Planning or Financial Management application with ISO currency codes.) Then, perform any currency conversions with those rates, as nothing is recalculated as part of this process.

Any data that is coming in with the default currency of the application is loaded to locale.

 Note:

Intersection checks are not performed on exchange rates. In addition, you cannot drill through on exchange rates.

Defining Metadata Rules

You can create metadata rules once and rerun the rules as necessary.

For general ledger source systems:

- For Fusion and E-Business Suite source systems, the chart of accounts is the collection of general ledger segments with various value sets, which are mapped to the dimensions to pull the dimension members and hierarchies.
- Similarly for PeopleSoft Enterprise Financial Management, the chartfields are mapped to the dimensions to pull the dimension members and hierarchies.

 Note:

Metadata rules are not used in Oracle Hyperion Financial Data Quality Management, Enterprise Edition integrations with human resources source systems.

Before you define metadata rules:

- Ensure that your source system data does not include special characters, which are not supported in Oracle Hyperion Financial Management target applications.
- Register your source systems and target applications for use with FDMEE. See [Registering Enterprise Resource Planning \(ERP\) Source Systems](#) and [Registering Target Applications](#).
- Select the source accounting entities in the registered source system. See [Selecting Source Accounting Entities](#).
- Define the import format. See [Working with Import Formats](#).
- Define the location. See [Defining Locations](#).

 Note:

Oracle Hyperion EPM Architect supports Shared and Local dimensions. FDMEE also supports Shared and Local dimensions in applications.

At a high level, follow this process to define metadata rules:

1. Create the metadata rule.

2. Select the dimension.
3. Define the mapping details and determine how you want to handle intercompany segments.
4. Define dimension attributes. See [Defining Dimension Attributes](#).
5. **Optional:** Define the segment hierarchies or chartfield trees to be extracted.
6. Save and run the metadata rule.
7. **Optional:** Check the status of the rule. See [Viewing Process Details](#).

To create metadata rules:

1. On the **Workflow** tab, under **Metadata**, select **Metadata Rule**.

 Note:

You cannot create multiple metadata rules for the same ledger or business unit for each target application.

2. From the **POV** bar, select the location to use for the metadata rule.
3. Click **Add**.
A blank line is displayed at the top of the Dimension Mappings summary grid.
4. In the **Mapping** details area, from **Dimension**, select the dimension.
The dimensions listed are based on the import format.
When a Dimension is selected, the Dimension Classification field prefills.
5. Define the mapping details for each dimension that you select.
6. Repeat steps 4-5 for each dimension.
7. Click **Save**.

Defining the Metadata Rule Details

Single segment or chartfield mappings define a simple one-to-one mapping between source and target dimension members. When you create single segment or chartfield mappings, you can optionally define:

- A member prefix or suffix type and value
- Segment hierarchies to extract from the source system
- Orphan member handling
- Statistical Account handling

To define the mapping details:

1. **Optional:** In **Prefix/Suffix Type**, select **Prefix** or **Suffix**.
2. In **Prefix/Suffix Value**, enter the member prefix or suffix.

Member prefixes are inserted before the source member code. Although optional, it is important to prefix the segment values with a prefix defined for the corresponding dimension when those members do not exist in the target application.

Member suffixes are inserted after the source member code.

 Note:

When you perform the next steps to define the hierarchy region starting parent, consider that Oracle Hyperion Planning, Oracle Essbase, and Oracle Hyperion Financial Management do not allow members to roll up to the parent under the same root. When extracting, specify hierarchies where every node has one parent. Fusion and E-Business Suite support instances where a segment value can roll up to two parents.

3. Select **Concatenate Name to Alias** to concatenate the Name and Alias.
4. From **EPMA Option**, select the EPMA metadata load option.

Metadata load options:

- **Merge as Primary**—Merge as Primary processes all sections and adds new dimensions, members, relationships, properties and associations that exist in the source, but do not exist in the Shared Library or target application. No dimensions, members, relationships, properties, or associations are removed. If a dimension, relationship, member, or property specified in the source exists in the Shared Library or target application it is overwritten with the value specified in the source. (Members are not moved, but overwritten.) Properties not included in the source are unchanged in the Shared Library or target application.

 Note:

If a member has multiple instances in a dimension, the IsPrimary column specifies which instance is primary and which instances are shared. The IsPrimary parameter is optional; however, Oracle strongly recommend that you use it. If left undefined, it defaults to "True," which assumes that the member is primary. If you do not define the IsPrimary parameter or when there are duplicate members defined as primary, a warning is displayed in the Import Results file.

- **Merge as Move**—This mode processes only members with IsPrimary set to true, and ignores any others. During processing, the first line specifying a primary location for a member is executed, any subsequent lines specifying a different primary location are ignored, and a warning message is displayed.
- **Replace**—All new elements are added and all property updates are made. Then, any members or member relationships that are not specified in the source are deleted from the Shared Library or target application. You can also use replace mode to reorder members under a parent. Properties not included in the source are unchanged in the Shared Library or target application.

This mode does not create Shared members unless they are defined in the import file (IsPrimary=false). This mode can detect a primary member under a new parent and process it as a move.

 Note:

The Replace option does not display the Reorder Type and Reorder Existing Members options.

5. For Fusion and E-Business Suite source systems:

- a. For Planning and Essbase applications—Account dimension mapping details. Select the **Time Balance Property for Balance Sheet Accounts** and **Time Balance Property for Income Statement Accounts**.

The time balance property specifies how the value of summary time periods is calculated. If set to Flow, it is an aggregate of all values for a summary time period as a period total. If the time balance property is set to "Balance," it is considered an ending value in a summary time period for the period total.

- b. From the **Hierarchy Region** tab, click **Add** to define the hierarchy region.

- c. Click  or enter the Starting Parent.

- d. Select **Base Hierarchy**.

The base hierarchy indicates the part of the hierarchy that is the base, and those parts of the hierarchy that share the same parent. All nonshared members of base hierarchies have the "Store Data" property set. However, shared members cannot be set to "Store Data."

- e. Select a prefix or suffix for the hierarchy, and then enter a prefix/suffix value.

The parent prefix is applied only to the parent. To enable alternate rollup hierarchies, the Prefix/Suffix value applies only to parent members. Parent members cannot be shared and must have a unique name. In addition, parent members do not store data.

6. For PeopleSoft source systems:

- a. From the **Hierarchy Region** tab, click **Add** to define the hierarchy region tree.

- b. In **Tree**, enter the tree name for the hierarchical structure.

Trees depict hierarchical structures that represent a group of summarization rules for a particular database field. For example, a tree can specify how your manufacturing locations should be summarized, or rolled up, for reporting purposes. A tree can also show the reporting relationships within an organization by specifying which individual department should be summarized into territories, territories into regions, and regions into countries. Similarly, a tree can categorize items in a catalog.

- c. in **Effective Date**, specify the effective date of the tree.

Using effective dates with trees enables you to specify new objects, departments, reporting relationships, or organizational structures in advance and have them take effect automatically. You can also use trees with past, present, or future effective dates when reporting on current or historic data.

- d. Click  or enter the Starting Parent.

- e. Select **Base Hierarchy**.

In a base hierarchy, other parts of the hierarchy that share a parent are shared. They cannot be set to "Store Data." Unshared members; however,

have the "Store Data" property set. All nonshared members of base hierarchies have the "Store Data" property set. However, shared members cannot be set to "Store Data."

- f. Select a prefix or suffix for the hierarchy, and then enter a prefix/suffix value.

The parent prefix is applied only to the parent. To enable alternate rollup hierarchies, the Prefix/Suffix value applies only to parent members. Parent members cannot be shared and must have a unique name. In addition, parent members do not store data.

- g. From **Select how to process source orphan members**, select the method for handling orphan members:

- Ignore—No orphan members from the source are extracted.
- Create as Root Member—Root members are created, and orphan members are not. All members are created at the top level of the hierarchy.
- Create as Children of—Orphan members are placed as children of the member specified in the entry field to the right "Create as Children of field."

7. Planning only: Select the **Plan Type** for the Accounting and Entity dimensions.
8. Financial Management only: If you are mapping an Entity dimension, enter the following details for intercompany segments, depending on your source system:

- Intercompany Segment Value
- For Intercompany Default, select **Yes** or **No**.

In the scenario where the Intercompany segment is defined, the Intercompany transactions are identified based on the Intercompany segment. Typically, one set of natural accounts is required for accounting the receivable and payable among the Intercompanies.

In the scenario where there is no Intercompany segment, identify the Intercompany transactions based explicitly on the natural account segment. Typically, all combinations of receivable and payable accounts exist among the transacting partner companies. Only with these natural accounts are the receivable and payable positions between the Intercompanies known.

9. Click **Save**.

Defining Dimension Attributes

The Attribute tab provides a table with a list of attributes for dimension. For each attribute, you can specify a default value. Note that the attributes vary by application type and dimension as shown below.

Note:

The default values are not validated. Refer to your application documentation for valid values.

Table 1-6 Dimension Attributes by Application Type

Application Type	Dimension	Attribute
Planning	Account	Time Balance for Balance Sheet Time Balance for Income Statement Data Storage Parent (sets the default for a parent node that you want to make different from the child node. In some cases, the child and parent data storage attributes are not the same, and this lets you specify the appropriate default for this attribute for a parent.) Data Storage Expense Reporting Account Type
HFM	Account	Consolidation Account Type Custom 1 Top Member Custom 2 Top Member Custom 3 Top Member Custom 4 Top Member
HFM	Entity	IsICP Currency
Essbase	Account	

To specify metadata attributes for a Peoplesoft source system:

1. From the **Attribute** tab, click **Add**.
2. In **Time Balance for Balance Sheet**, **Time Balance for Income Statement**, and **Expense Reporting**, specify how the value of the summary time periods is calculated.

"Flow" it is an aggregate of all values for a summary time period as a period total. If the time balance property is set to "Balance," then it is considered an ending value in a summary time period for the period total.

To use the system default, click **Use System Default**.

To specify a custom default value:

1. Select the **Attribute** tab.
2. Enter the default value for the attribute.
3. To use the custom default value, clear the **Use System Default** field.
4. Click **Save**.

Managing Metadata Rules

You can perform the following tasks:

- Edit metadata rules—See [Editing Metadata Rules](#).
- Run metadata rules—See [Running Metadata Rules](#).

- Check the metadata rule process details—See [Deleting Metadata Rules](#).
- Delete dimension mappings or hierarchies in metadata rules—See [Checking the Metadata Rule Status](#).

Editing Metadata Rules

If the metadata rule is not running, you can modify the rule.

To edit metadata rules:

1. On the **Workflow** tab, under **Metadata**, select **Location**.
2. Enter the **Location Name** or click  to select the location.
3. Add or modify the dimension mappings or mapping details as necessary.
4. Click **Save**.

Running Metadata Rules

You can run the metadata rule to load updates and push the metadata into the target application. All submitted rules are processed by Oracle Data Integrator.

To submit the metadata rule:

1. On the **Workflow** tab, under **Metadata**, select **Metadata Rule**.
2. In **Metadata**, enter the **Location Name** or click  to select the location.
3. Select the metadata rule.
4. Click **Execute**, and then click **OK**.

Checking the Metadata Rule Status

After you run a metadata rule, you can check the status on the Process Details page. You can click the Status icon on the Metadata page to link to the Process Details page and view the process details. See [Viewing Process Details](#).

Tip:

You can also check the status of the rule in Oracle Data Integrator.

Deleting Metadata Rules

You can delete dimension mappings or hierarchies in the mapping details for metadata rules created in Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

To delete dimension mappings or hierarchies in a metadata rule:

1. On the **Workflow** tab, under **Metadata**, select **Metadata Rule**.
2. In **Metadata**, enter the **Location Name** or click  to select the location.
3. Select the row in the Dimension Mappings or Hierarchy Region area.

4. Click **Delete**.

Loading Source System Hierarchies into EPM Dimensions

Metadata within the source system changes over time, as well as the metadata and hierarchies in the target system.

The management of hierarchies is an ongoing process, with frequent changes due to updates in business functions and organizations. When managing hierarchies between source and target systems, users generally create new hierarchies, replace old hierarchies with new hierarchies or update hierarchies.

Managing hierarchies between systems becomes difficult because of the size of the hierarchies, the latency between system updates, and the needs of operational systems versus analytical systems. When managing hierarchies as part of the general ledger integration process, consider:

- The only operations between hierarchy management in the source system and target application are creating and updating the hierarchies by merging in the target. Oracle Hyperion Financial Data Quality Management, Enterprise Edition never deletes hierarchies or members in a target application. If additional members or hierarchies are not specified, FDMEE ignores them.
- When you integrate a hierarchy from the source system to the target system, select the node from the source that serves as the root node in the target.
- The integration pushes the hierarchy into the target system and reports any errors encountered during the process.

You can use the Process Details page to view errors logged in FDMEE. You can also select the Log link to review the Oracle Data Integrator log file. For Oracle Hyperion EPM Architect applications, you can also view profile creation errors in the Job Console. See [Viewing Process Details](#) or the appropriate product documentation for additional information.

- A hierarchy selection in the dimension mapping is optional; however, you must at least determine how to handle members not in a hierarchy. For example, you can create children of a selected node as orphans, or you can choose not to carry over orphans. (This option applies only to Performance Management Architect).

Extracting Human Resource Data

Follow this process to extract metadata and data from human resource source systems into Oracle Hyperion Public Sector Planning and Budgeting target applications:

1. Register source systems in Oracle Hyperion Financial Data Quality Management, Enterprise Edition by adding details for Oracle Data Integrator and FDMEE, which are specific to the PeopleSoft Human Capital Management source system.
2. Register your target application.
3. Select the Source Accounting Entities (Business Units).
4. Create the human resource data load rules.
5. Run human resource data rules to extract data and/or metadata from the source system and push it into the target applications.

Data and metadata are staged in the FDMEE staging tables, extracted from the PeopleSoft Human Capital Management source system, and then loaded into the target application.

Navigating FDMEE

From Oracle Hyperion Enterprise Performance Management Workspace, you can access Oracle Hyperion Financial Data Quality Management, Enterprise Edition from the Navigate menu. (**Navigate, Administer, Data Management**)

Toolbars

Help Menu

Use the Help menu to access Oracle Hyperion Financial Data Quality Management, Enterprise Edition online help, Oracle technical support, the EPM documentation located on the Oracle Technology Network, Oracle website, and information about FDMEE.

Task Pane Options

The Tasks pane is a resizeable window to the left of Oracle Hyperion Financial Data Quality Management, Enterprise Edition Workspace. It provides easy access to FDMEE options and features. The Tasks pane consists of the Workflow and Setup tabs.

Workflow Tasks

From the Workflow tab, you can integrate metadata and data from an Enterprise Resource Planning (ERP) source system into an Enterprise Performance Management (EPM) target application:

You can also load data from a file and other source systems.

- Data Load
 - Data Load Workbench
 - Data Load Rule
 - Data Load Mapping
- Metadata—Metadata Rule
- HR Data Load—HR Data Load Rule
- Other
 - Batch Execution
 - Report Execution
 - Script Execution
- Monitor—Process Details

Setup Tasks

From the Setup tab you can administer source and target systems, specify report and batch definitions, and manage application settings.

Available tasks:

- Configure
 - System Settings
 - Application Settings
 - Security Settings
 - User Settings
- Register
 - Source System
 - Target Application
 - Source Accounting Entity
 - Source Adapter
- Integration Setup
 - Import Format
 - Location
 - Period Mapping
 - Category Mapping
 - Excel Interface
- Data Load Setup
 - Logic Group
 - Check Rule Group
 - Check Entity Group
- Scripts
 - Script Editor
 - Script Registration
- Reports
 - Query Definition
 - Report Definition
- Batch—Batch Definition

Working with Data in Grids

Most screens display data in one or more grids. To manipulate grid data, perform one or more actions:

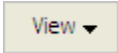
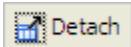
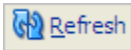
- To add a record, click **Add**.

- To delete, select a record, and then click **Delete**.
- To delete all records in a grid, click **Delete All**.
- To edit a record, click within its cell, and start typing. When applicable, you can also select the value to edit, and then click .
- To search items in a column, enter the search value in the blank field above the column of the value, and then press **Enter**. If the value is matched, it is displayed as the first item.
- To cancel all changes made to a row, select the row, and then click **Cancel**.
- To save all changes made to a row, select **Save**.

FDMEE User Interface Elements

The following elements are common on Oracle Hyperion Financial Data Quality Management, Enterprise Edition pages.

Table 1-7 Elements Common on FDMEE Pages

Button	Description
	Customize your view. Options include: • Columns—You can choose “Show All” to display all columns or choose individual columns to display. • Detach—Use to detach the column grid. When you detach the grid, the columns display in their own window. To return to the default view, select View, and then click Attach or click Close. • Reorder Columns—Use to change the order of the columns that are displayed. You can select a column, and then use the buttons on the right to change the column order.
	Use to detach the column grid. When you detach the grid, the columns are displayed in their own window. To return to the default view, select View , and then click Attach or click Close .
	Refreshes the data. For example, if you submit a rule, refresh to see if the status changes from Running to Complete.

 Note:

Refresh does not display on the FDMEE setup screens.

Table 1-7 (Cont.) Elements Common on FDMEE Pages

Button	Description
	Use to toggle the filter row. You can use the filter row to enter text to filter the rows that are displayed for a specific column. You can enter text to filter on, if available, for a specific column, and then press [Enter]. For example, on the Process Details page, to view only processes for a specific location, enter the name of the location in the Location text box. The Query by Example button displays on the following FDMEE setup screens: Target Application, Source Accounting Entities, Import Format, Location, Data Load Workbench, and Process Details. To clear a filter, remove the text to filter by in the text box, and then press [Enter]. All text is case sensitive.
	Use to select an artifact on a page, such as a target application, member, or general ledger responsibility. When you click the Search button, the Search and Select dialog box is displayed. In some cases available advanced search options enable you to enter additional search conditions. See Advanced Search Options.

Advanced Search Options

The Search button is common to many Oracle Hyperion Financial Data Quality Management, Enterprise Edition pages. When you select the Search button, if the Advanced Search button is available, you can enter additional search conditions. The fields that are displayed in the advanced search options differ depending on what artifact you are selecting. The following operators are supported:

- Starts with
- Ends with
- Equals
- Does not equal
- Less than
- Greater than
- Less than or equal to
- Greater than or equal to
- Between
- Not between
- Contains

- Does not contain
- Is blank
- Is not blank

Using the POV Bar

For the Data Load Workbench, the POV bar shows the current:

- Location
- Period
- Category
- Data Rule

Location **KS7DIM_EBSTB** Period **Jan-08** Category **Actual** Rule **KS7DIM_EBSTB** Source **File** Target **KS7DIM**

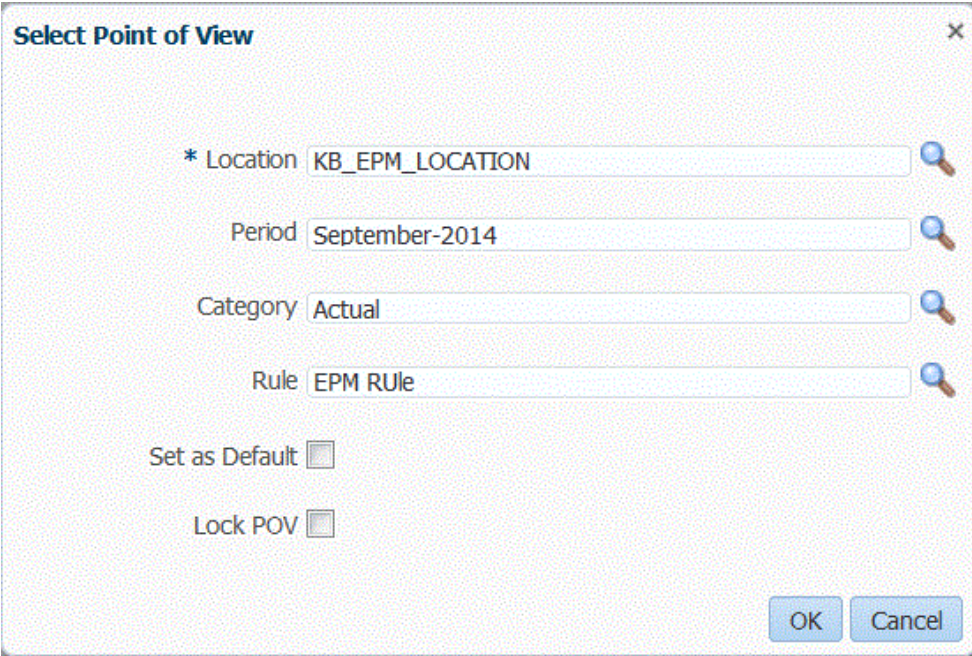
By default, only the data rule assigned to the Category POV is displayed.

The Source System and Target Application are displayed as context information.

Selecting the Location POV

To select another Location POV:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Workbench**.
2. From the **POV** bar, double-click the **Location** field.



The "Select Point of View" dialog box is shown. It contains four text input fields with search icons to their right: "Location" (containing "KB_EPM_LOCATION"), "Period" (containing "September-2014"), "Category" (containing "Actual"), and "Rule" (containing "EPM RULE"). Below these fields are two checkboxes: "Set as Default" and "Lock POV", both of which are currently unchecked. At the bottom right of the dialog are "OK" and "Cancel" buttons.

3. In **Select Point of View**, in **Location**, enter a full or partial string for the new location, and then click **OK**.

4. **Optional:** To search on another location, from the **Location** drop-down, click **More**, navigate to the location on the **Search and Select: Location** screen, and then click **OK**.
5. **Optional:** In **Select Point of View**, select **Set as Default** to use the new location as the default location.

When a POV selection is set as a default, the user profile is updated with the default selection.
6. Click **OK**.

Setting the Period POV

The Oracle Hyperion Financial Data Quality Management, Enterprise Edition administrator controls which accounting period is active for all users. This feature prevents users from inadvertently loading data into incorrect periods. When you log on to FDMEE, the application identifies the global period value and automatically sets the POV to the current value.

To select another Period POV:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Workbench**.
2. From the **POV** bar, double-click the **Location** field.
3. From **Select Point of View**, in **Period**, enter a full or partial string for the new period, and then click **OK**.
4. **Optional:** To search on another period, from the **Period** drop-down, click **More**, navigate to the period on the **Search and Select: period** screen, and then click **OK**.
5. **Optional:** In **Select Point of View**, select **Set as Default** to use the new period as the default period.

When a new POV selection is set as a default, the user profile is updated with the default selection.
6. Click **OK**.

Setting the Category POV

The Oracle Hyperion Financial Data Quality Management, Enterprise Edition administrator controls the data category active for all users. This feature prevents users from inadvertently loading data to incorrect categories.

To select another Category POV:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Workbench**.
2. From the **POV** bar, double-click the **Location** field.
3. In **Select Point of View**, in **Category**, select the new category, and then click **OK**.
4. **Optional:** In **Rule**, select the rule assigned to the Category POV.
5. Select **Set as Default** to use the new category as the default category.

When a POV is set as a default, the user profile is updated with the default selection.
6. Click **OK**.

Locking and Unlocking a POV

Locking the POV prevents user from modifying the data. When a location has been locked for a particular period or category, users cannot import, validate, export, or rerun the validation report.

When a location is locked, a lock symbol () is displayed in the POV bar.

Data in a locked POV can only be loaded when the locked POV is "unlocked."

The POV Lock options include:

- Lock POV
- Unlock POV
- Lock All Locations

(The Lock All Locations and Unlock all Locations features is available only to administrators on the Lock POV for All Locations screen. See [Locking and Unlocking All \(POV\) Locations](#).

- Unlock All Locations

The POV lock is referenced in:

- Data Load Workbench
- Data Load Rule
- Batch Execution

To lock a POV:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Workbench**.

The POV bar is also displayed on the Data Load Rule screen and Batch Execution screen

2. On the **POV** bar, double-click the **Location**.

3. In **Select Point of View**, click **Lock POV**.

(The Lock POV and Unlock POV options are only available to administrators using the **Allow Unlock by Location** option on the Lock POV for All Locations screen. See [Locking and Unlocking All \(POV\) Locations](#).

The message: "Are you sure you want to lock selected POV? Data cannot be loaded to a locked POV." is displayed.

4. From **Lock POV Confirmation**, click **OK**.

An informational message shows that the POV is locked.

5. **Optional:** To unlock a locked POV:

- a. From the **POV** bar, double-click the **Location** field.
- b. From **Select Point of View**, click **Unlock POV**.
- c. Click **OK**.

An informational message shows that the POV has been successfully unlocked.

Administration Tasks

These sections describe how to perform key operations in Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

Predefining a List of Profiles

Oracle Hyperion Financial Data Quality Management, Enterprise Edition uses a predefined list of profiles. You can define values for these profiles to accommodate various business needs. Profiles can be set at the following levels:

- System (applies to the entire system)
- Application (applies to specific target application)
- User (applies to a specific user)
- Security (Role, Report, Batch, Custom Script, and Location)

Setting System-Level Profiles

Use the System Settings Option to update or clear System level profiles that apply to entire system.

To define system settings:

1. On the **Setup** tab, under **Configure**, select **System Settings**.
2. In **System Settings**, in **Profile Type**, select the specific profile to list on the System Settings screen.

Available profile types:

- All
- File—In addition to file-specific system settings, selecting the File profile type displays the "Create Application Folders" button. This feature instructs the system to create a folder structure in the path specified in the Application Root Directory setting.
- ODI—Use to set ODI password and repository connection information. When this information has been added or changed, you can click **Check ODI Connection** to view if connection to the ODI agent was made successfully.
- Other—Use to set profiles associated with an EPMA data source, User Language, User Interface Theme, and Default Check Report.
- Point-of-View

The profile type that you select determines the settings that you can add or modify on the screen.

3. Select the option and add the new value in **Value**.



If  is displayed in the Select field, you can search on the value.

**Note:**

When you install and configure Oracle Hyperion Financial Data Quality Management, Enterprise Edition, Oracle Data Integrator is automatically installed and configured for you. The database for Oracle Data Integrator is in the same database as FDMEE and the Oracle Data Integrator agent deployed to the FDMEE Managed Server. You should be familiar with Oracle Data Integrator and review the Oracle Data Integrator documentation set before specifying or changing the installation defaults.

4. Click **Save**.

Table 1-8 System Setting Profile Options

Profile Type	Profile Option	Profile Description
All	Includes all profile types	

Table 1-8 (Cont.) System Setting Profile Options

Profile Type	Profile Option	Profile Description
File	Application Root Folder	The Application Root Directory must be set up on the server, specified in this field, and the Create Application Folder option must be executed as first steps when using FDMEE. The Application Root folder identifies the root directory of the FDMEE application. This folder is located on the FDMEE server and functions as the root folder for all FDMEE activities. Based on this parameter, FDMEE saves log files, generated files and reports to the appropriate folder under this root directory. Parameters must be set up on the server separately from this setup step. When you select the File profile type, the System Setting screen displays the Create Application Folder button. This feature instructs the system to create a folder structure in the path specified in this field. The folder structure is (with sub-folders in each): - data - inbox - outbox Within the inbox, locations are created when they are added in the Locations option. See also FDMEE Application Folder Architecture. When you specify a folder at the application level, and select the Create Application Folders option, a set of folders is created for the application that includes a scripts folder. Create scripts specific to an application in this folder. This is especially important for event scripts that are different between applications. If you do not set up an application level folder, then you cannot have different event scripts by application. If you specify a Universal Naming Convention (UNC) path, share permissions on the folder must allow access to the DCOM user for read/write operations. Use a Universal Naming Convention (UNC) path for the application root folder when Oracle Hyperion Financial

Table 1-8 (Cont.) System Setting Profile Options

Profile Type	Profile Option	Profile Description
		Management and FDMEE are on separate servers. Contact your server administrator to define the required UNC definition. If an UNC path is not entered, then you must enter the absolute path. For example, specify <i>C:\Win-Ovu31e2bfiefdmee</i>
File	Create Location Folder	This setting instructs the system to create a location folder in the inbox when a location is created. Available values are Yes or No. Set this option once and do not change it. This setting is optional but recommended.
File	Archive Mode	Specifies whether archived files are copied or moved to the archive location. Enter Copy, Move, or None. If you select Copy, then the file is left in the inbox. If you select Move, then the file is copied to the archive folder and deleted from the inbox. The folder named data is the archive folder. When the file is moved to the archive location, it is renamed as follows: <Process ID><Year><Month><Day>.<Original Extension> For example, if the source file name is <i>BigFile.csv</i>, and it was loaded for period Mar-07 with a period key of 03/01/2007, and if the process id was 983, then the resulting file name is <i>98320070301.csv</i>.
File	Excluded File Upload Wild Cards	Specify file extensions that cannot be uploaded. Enter *.* to disallow all file uploads.

Table 1-8 (Cont.) System Setting Profile Options

Profile Type	Profile Option	Profile Description
File	Batch Size	Specify the number of rows read at a time from the file to memory. This parameter is mainly used for performance. When data is loaded, this setting determines how many records are stored in the cache. For example, when 1000 is specified; the system stores 1,000 records in cache. Similarly, when 5000 is specified, the system stores 5,000 records in cache and commit. Determine this setting by Server Memory, and adjust as needed.
File	File Character Set	Specify the method for mapping bit combinations to characters for creating, storing, and displaying text. Each encoding has a name; for example, UTF-8. Within an encoding, each character maps to a specific bit combination; for example, in UTF-8, uppercase A maps to HEX41. Click  to view available character sets. Encoding refers to mapping <i>bit</i> combinations to characters for creating, storing, and displaying text. Convert the encoding to UNICODE if your source file is not in one of the supported formats.
File	Encrypted Password Folder	Specify the directory where the files that store passwords in encrypted form is located. This encrypted password folder is used with the "Update Configuration File" button. See Working with Batch Scripts.
ODI	ODI User Name	Specifies the Oracle Data Integrator user name used to access the Oracle Data Integrator master repository. For example, enter Supervisor. This setting is defined automatically when ODI is configured, but you can customize it if necessary.

Table 1-8 (Cont.) System Setting Profile Options

Profile Type	Profile Option	Profile Description
ODI	ODI Password	Specifies the Oracle Data Integrator database schema used to access the Oracle Data Integrator master repository. For example, enter Master. This setting is defined automatically when ODI is configured, but you can customize it if necessary.
ODI	ODI Execution Repository	Specifies the repository where all scenarios are stored. For example, enter ERPI_REF. This setting is defined automatically when ODI is configured, but you can customize it if necessary.
ODI	ODI Work Repository	Specifies the (execution) repository which contains runtime objects (for example, scenarios). The work repository can be linked with only one Master Repository. For example, enter FDMEE_WORK_REF.
ODI	ODI Master Repository Driver	Specifies the driver of the ODI master repository. This setting is defined automatically when ODI is configured, but you can customize it if necessary.
ODI	ODI Master Repository URL	Specifies the URL of the server where the Oracle Data Integrator master repository is installed. This setting is defined automatically when ODI is configured, but it can be customized if necessary. For example, enter <code>jdbc:oracle:thin:@serverdatabase.oracle.com:1521:orcl</code>.
ODI	ODI Master Repository User	Specifies the Oracle Data Integrator master repository user name. This setting is defined automatically when ODI is configured, but you can customize it if necessary.
ODI	ODI Master Repository Password	Specifies the Oracle Data Integrator master repository password. This setting is defined automatically when ODI is configured, but you can customize it if necessary.
Other	EPMA Data Source Name	Specifies the EPMA Data Source name for the EPMA Interface Data Source.

Table 1-8 (Cont.) System Setting Profile Options

Profile Type	Profile Option	Profile Description
Other	User Language	Specify the system default language of the user version of FDMEE. FDMEE uses the user language to query the language data, for example, column titles, segment name, and so on.
Other	User Interface Theme	The default theme contains all the colors, styles, and general-use icons that are displayed in the user interface. FDMEE uses BLAF+ as the default value.
Other	Default Check Report	Specify the type of Check Report to use as the default check report. The following are pre-seeded check reports, but you can create a new one and specify it here: • Check Report—displays the results of the validation rules for the current location (pass or fail status). • Check Report Period Range (Cat, Start per, End per)—Displays the results of the validation rules for a category and selected periods. • Check Report by Val. Entity Seq.—Displays the results of the validation rules for the current location (pass or fail status); sorted by the sequence defined in the validation entity group. • Check Report with Warnings—Displays the results of the validation rules for the current location. Warnings are recorded in validation rules and shown if warning criteria are met. This report does not show rules that passed the validation.
Other	Default Intersection Report	Specify the type of Intersection Check Report to use as the default intersection check report at the system level. Intersection reports identify data load errors and are generated as part of the data validation step in the Data Load Workbench. The reports are available in two formats: <i>Dynamic Column</i> or <i>Fixed Column</i> format. The Fixed Column displays up to four custom dimensions.

Table 1-8 (Cont.) System Setting Profile Options

Profile Type	Profile Option	Profile Description
Other	Batch Timeout in Minutes	When a batch job is run in sync mode (immediate processing), specify the maximum time the job can run. In sync mode, FDMEE waits for the job to complete before returning control.
Other	Enable Event Script Execution	Select Yes to enable the execution of application events such as before loading data (BefLoad) or after validation (AftValidate). Select No to disable the execution of application events.
Other	SQL Server Database Provider	Specify the name of the SQL Server database provider. Available SQL Server database providers: • SQLOLEDB • SQLNCLI10 (SQL Server 2008) • "Need Name" (SQL Server 2012)
Other	Log Level	Specify the level of detail displayed in the logs. A log level of 1 shows the least detail. A log level of 5 shows the most detail. Logs are displayed in Process Details by selecting the Log link.
Other	Check Report Precision	Specify the total number of decimal digits for rounding numbers, where the most important digit is the left-most nonzero digit, and the least important digit is the right-most known digit.
Other	Display Data Export Option "Override All Data"	Specify Yes to display the Override All Data option in the Export Mode drop-down located on the Execute Rule screen. When you select to override all data, the following message is displayed "Warning: Override All Data option will clear data for the entire application. This is not limited to the current Point of View. Do really want to perform this action."
Other	Enable Map Audit	Set to Yes to create audit records for the Map Monitor reports (Map Monitor for Location, and Map Monitor for User). The default value for this setting is No .

Table 1-8 (Cont.) System Setting Profile Options

Profile Type	Profile Option	Profile Description
Other	Access to Open Source Document	When drilling down to the FDMEE landing page, this setting determines access to the Open Source Document link (which opens the entire file that was used to load data). - Administrator—Access to Open Source Document link is restricted to the administrator user. - All Users—Access to the Open Source Document link is available to all users. All Users is the default setting.
Other	Map Export Delimiter	Sets the column delimiter value when exporting member mappings. Available delimiters are: ! (exclamation mark) , (comma) ; (semi-colon) (pipe)
POV	Default POV Location	Specifies the default POV location. These preferences take precedence when no equivalent settings are in Application Settings or User Settings.
POV	Default POV Period	Specifies the default POV Period. These preferences take precedence when no equivalent settings are in Application Settings or User Settings.
POV	Default POV Category	Specifies the default POV Category. These preferences take precedence when no equivalent settings are in Application Settings or User Settings.
POV	Global POV Mode	When this is set to Yes, other POVs (Application Level and User Level POVs) are ignored.

Setting Application-Level Profiles

Use the Application Settings option to update or clear application-level profiles that apply to the target application.

To set an application level profile:

1. On the **Setup** tab, under **Configure**, select **Application Settings**.
2. In **Application Settings**, in the **Target Application** drop-down, select the target application to which the application profile applies.
3. Select application level profile settings.
4. **Optional:** To clear a setting, select the value, and then click **Delete**.

The value is removed but is deleted only when you save it.

5. Click **Save**.

Table 1-9 Application Level Profile Options

Option	Description
Application Root Folder	The Application Root folder is the root folder for storing all files used to load data to this EPM application. You can use a separate root folder for each EPM application. Based on this parameter, Oracle Hyperion Financial Data Quality Management, Enterprise Edition saves log files, generated files and reports to the appropriate folder under this root directory. Parameters must be set up on the server separately from this setup step. Selecting the Create Application Folder button instructs the system to create a folder structure in the path specified in this field. The folder structure is (with sub-folders in each): - data - inbox - outbox When you specify a folder at the application level, and select the Create Application Folder option, a set of folders is created for the application that includes a scripts folder. Create scripts specific to an application in this folder. This is especially important for event scripts that are different between applications. If you do not set up an application level folder, then you cannot have different event scripts by application. If you specify a Universal Naming Convention (UNC) path, share permissions on the folder must allow access to the DCOM user for read/write operations. Use a Universal Naming Convention (UNC) path for the application root folder when Oracle Hyperion Financial Management and FDMEE are on separate servers. Contact your server administrator to define the required UNC definition. If an UNC path is not entered, then you must enter the absolute path. For example, specify <code>C:\Win-Ovu31e2bfie\fdmee</code>

Table 1-9 (Cont.) Application Level Profile Options

Option	Description
File Character Set	Specify the method for mapping bit combinations to characters for creating, storing, and displaying text. Each encoding has a name; for example, UTF-8. Within an encoding, each character maps to a specific bit combination; for example, in UTF-8, uppercase A maps to HEX41. Click  to view available character sets. Encoding refers to mapping <i>bit</i> combinations to characters for creating, storing, and displaying text. Convert the encoding to UNICODE if your source file is not in one of the supported formats.
Default POV Location	Specify the default POV location.
Default POV Period	Specify the default POV Period.
Default POV Category	Specify the default POV Category.
User Language	Specify the application default language of the user version of FDMEE.
User Interface Theme	Set the Oracle design pattern for the applications. FDMEE uses BLAF+ as the default user interface value.
Default Intersection Report	Specify the type of Intersection Check Report to use as the default intersection check report at the application level. Intersection Check reports identify data load errors and are generated as part of the data validation step in the Data Load Workbench. The reports are available in two formats: <i>Dynamic Column</i> or <i>Fixed Column</i> format. The Fixed Column displays up to four custom dimensions.

Table 1-9 (Cont.) Application Level Profile Options

Option	Description
Default Check Report	Specify the type of Report to use as the default report at the application level. The following are pre-seeded reports, but you can create a new one and specify it here: • Check Report—Displays the results of the validation rules for the current location (pass or fail status). • Check Report Period Range (Cat, Start per, End per)—Displays the results of the validation rules for a category and selected periods. • Check Report by Val. Entity Seq.—Displays the results of the validation rules for the current location (pass or fail status); sorted by the sequence defined in the validation entity group. • Check Report with Warnings—Displays the results of the validation rules for the current location. Warnings are recorded in validation rules and shown if warning criteria are met. This report does not show rules that passed the validation.
Enable Event Script Execution	Select Yes to enable the execution of application events such as before loading data (BefLoad) or after validation (AftValidate). Select No to disable the execution of application events.
Log Level	Specify the level of detail displayed in the logs. A log level of 1 shows the least detail. A log level of 5 shows the most detail. Logs are displayed in Process Details by selecting the Log link.
Check Report Precision	Specify the total number of decimal digits for rounding numbers, where the most important digit is the left-most non-zero digit, and the least important digit is the right-most known digit.
Display Data Export Option "Override All Data"	Display the "Override All Data" option on the Export Mode drop-down on the Execute Rule screen. When you select to override all data, the following message is displayed "Warning: Override All Data option will clear data for the entire application. This is not limited to the current Point of View. Do really want to perform this action."
Enable Map Audit	Set to Yes to create audit records for the Map Monitor reports (Map Monitor for Location, and Map Monitor for User). The default value for this setting is No.

Table 1-9 (Cont.) Application Level Profile Options

Option	Description
Access to Open Source Document	When drilling down to the FDMEE landing page, this setting determines access to the Open Source Document link (which opens the entire file that was used to load data). - Administrator—Access to Open Source Document link is restricted to the administrator user. - All Users—Access to the Open Source Document link is available to all users. All Users is the default setting.
Map Export Delimiter	Sets the column delimiter value when exporting member mappings. Available delimiters are: ! (exclamation mark) , (comma) ; (semi-colon) (pipe)

Locking and Unlocking All (POV) Locations

The lock all locations feature prevents data from being loaded to a selected POV by locking all locations related to the current period and category for an entire target application. When a location has been locked, you cannot import, validate, export, or re-run the validation.

When a location is locked, a lock symbol () is displayed in the POV bar.

The Lock All Locations feature is referenced in the:

- Data Load Workbench
- Data Load Rule
- Batch Execution

An “Unlock All Location” option is also available so that you can unlock all locked location. You can provide an option on the Select Point of View screen that enables users to unlock a POV by location.

For information on locking and unlocking an individual POV, see [Locking and Unlocking a POV](#)

To lock all locations for a POV:

1. On the **Setup** tab, under **Configure**, select **Application Settings**.
2. In **Application Settings**, from the **Target Application** drop-down, select the target application to which the application profile applies.
3. Click **Lock All Locations**.
4. In **Period**, select the period to lock.
5. In **Category**, select the category to lock.

6. Click **Allow Unlock by Location** to provide the **Unlock POV** option on the Select Point of View screen.

If **Allow Unlock by Location** is disabled, then the **Unlock POV** and **Lock POV** fields are not displayed on the Select Point of View screen.

7. Click **OK**.

All locations for the selected target application are locked.

To unlock a POV for all locations:

1. On the **Setup** tab, under **Configure**, select **Application Settings**.
2. In **Application Settings**, from the **Target Application** drop-down, select the target application to which the application profile applies.
3. Click **Unlock All Locations**.
4. In **Period**, select the period to unlock.
5. In **Category**, select the category to unlock.
6. Click **OK**.

All locations for the selected target application are unlocked.

Setting User Level Profiles

Use the User Settings option to update or clear user level profiles that apply to the user.

Note:

When the Global mode is defined, then user level profiles for the POV are not applicable.

To set a user level profile:

1. On the **Setup** tab, under **Configure**, select **User Settings**.
2. In **User Setting**, select the options to add or modify.
3. **Optional:** To clear a setting, select the value and from your keyboard, and then click **Delete**.
The value is removed, but it is deleted only when you save it.
4. Click **Save**.

Table 1-10 User Level Profile Settings

Option	Description
File Character Set	Specify the method for mapping <i>bit</i> combinations to characters for creating, storing, and displaying text. Each encoding has a name; for example, UTF-8. Within an encoding, each character maps to a specific <i>bit</i> combination; for example, in UTF-8, uppercase A maps to HEX41. Click  to view available character sets on the Search and Select screen. Encoding refers to mapping <i>bit</i> combinations to characters for creating, storing, and displaying text. You should convert the encoding to UNICODE if your source file is not in one of the supported formats.
Default POV Location	Specify the default POV location.
Default POV Period	Specify the default POV Period.
Default POV Category	Specify the default POV Category.
User Language	Select the default language of the user version Oracle Hyperion Financial Data Quality Management, Enterprise Edition user interface.
User Interface Theme	Specify the default theme of the user version of the FDMEE user interface.
Default Check Report	Specify the type of Check Report to use as the default check report at the user level. The following are pre-seeded check reports, but you can create a new one and specify it here: • Check Report—Displays the results of the validation rules for the current location (pass or fail status). • Check Report Period Range (Cat, Start per, End per)—Displays the results of the validation rules for a category and selected periods. • Check Report by Val. Entity Seq.—Displays the results of the validation rules for the current location (pass or fail status); sorted by the sequence defined in the validation entity group. • Check Report with Warnings—Displays the results of the validation rules for the current location. Warnings are recorded in validation rules and shown if warning criteria are met. This does not show rules that passed the validation.

Table 1-10 (Cont.) User Level Profile Settings

Option	Description
Default Intersection Report	Specify the type of Intersection Check Report to use as the default intersection check report at the user level. Intersection reports identify data load errors and are generated as part of the data validation step in the Data Load Workbench. The reports are available in two formats: <i>Dynamic Column</i> or <i>Fixed Column</i> format. The Fixed Column displays up to four custom dimensions.
Log Level	Specify the level of detail displayed in the logs. A log level of 1 shows the least detail. A log level of 5 shows the most detail. Logs are displayed in Process Details by selecting the Log link.
Map Export Delimiter	Sets the column delimiter value when exporting member mappings. Available delimiters are: • ! (exclamation mark) • , (comma) • ; (semi-colon) • (pipe)

Setting Security Options

In Oracle Hyperion Financial Data Quality Management, Enterprise Edition, administrators can enable location almost any user interface and report feature. FDMEE supports five levels of security:

- Role level security—Controls access to components of the user interface that each user can access.
- Report security—Controls the reports that can be executed based on the report groups assigned to a role.
- Batch security—Controls the batches that can be executed based on the batch groups assigned to a role.
- Custom scripts security—Controls the custom scripts that can be executed based on the custom script groups assigned to a role.
- Location security—Controls access to locations.

Security levels apply to users. Role and Location security levels assigned to users are compared at runtime. If a user is assigned a level that is equal to the level assigned to the feature that the user is trying to access, the feature is available to the user.

Role Level Security

Oracle Hyperion Financial Data Quality Management, Enterprise Edition security enables you to customize user access to user interface functions using the concept of roles. Roles are permissions that grant user access to functions. In FDMEE, default roles are assigned to functions that aggregate and tailor specific requirements. After

the functions are assigned to a role, the corresponding role is mapped to users when provisioning users in Oracle Hyperion Shared Services. The process of granting roles to users is described in the *Oracle® Enterprise Performance Management System User and Role Security Guide*.

To add role level security:

1. On the **Setup** tab, under **Configure**, select **Security Settings**.
2. In **Security Setting**, select the **User Interface** tab.
3. In **Role**, select the role category to which to assign access.

A list of roles is described in [Table 1](#).

The role category determines the display of functions associated with the selected role.

4. In **Function**, select the function to assign to the role.
5. Click **Save**.

Table 1-11 Role and Descriptions

Role	Description
Administrator	Grants access to all FDMEE functions.
Create Integration	Creates FDMEE metadata and data load rules.
Run Integration	Runs FDMEE metadata and data rules and fills out runtime parameters. Can view transaction logs.
Drill Through	Controls whether you can drill to the FDMEE landing page, which controls drilling to the source system.
HR Integration	Runs Human Resource data rules and fills out runtime parameters. Can view transaction logs.
Intermediate 2-9	Roles for intermediate levels are defined by the administrator.

Defining User Interface Security

To add user security:

1. On the **Setup** tab, under **Configure**, select **Security Settings**.
2. Select the **Role Security** tab.
3. From **Role**, select the role to which to assign access.
4. Select the **User Interface** tab.
5. In **Function**, select the user interface function to which to assign user interface security.
6. Click **Save**.

Defining Report Security

To define report security, you assign reports of a selected type to a group (see [Adding Report Groups](#)). Next, you assign the report group to a role. The role has access to all reports in the groups at execution time.

To add report level security:

1. On the **Setup** tab, under **Configure**, select **Security Settings**.
2. In **Role**, select the role to which to assign the report security.
A list of roles is described in [Table 1](#).
3. Select the **Report** tab.
4. In **Report Group**, select the report group to which to assign report security.
5. Click **Save**.

Defining Batch Security

To define batch security, you assign batches of a selected type to a group (see [Adding a Batch Group](#)). Next you assign the batch group to a role. The role has access to all batches in the groups at execution time.

To add batch security:

1. On **Setup** tab, under **Configure**, select **Security Settings**.
2. From **Role**, select the role to which to assign batch security.
A list of roles is described in [Table 1](#).
3. Select the **Batch** tab.
4. In **Batch Group**, select the batch group to assign batch security.
5. Click **Save**.

Defining Custom Script Security

To define custom scrip security, you assign custom scripts of a selected type to a group (see [Adding a Custom Script Group](#)). Next you assign the custom scripts group to a role. The role has access to all custom scripts in the groups at execution time.

To add role level security:

1. On the **Setup** tab, under **Configure**, select **Security Settings**.
2. From **Role**, select the role to which to assign custom script security.
A list of roles is described in [Table 1](#).
3. Select the **Custom Script** tab.
4. From **Custom Script Group**, select the custom script group to which to assign custom script security.
5. Click **Save**.

Defining Location Security

Location security (user access to locations) for Oracle Hyperion Financial Data Quality Management, Enterprise Edition is configured and enforced by options on the Location Security Settings tab. You define the user groups to create for each location. When a location is created or updated, then you can create as many groups as defined in the system settings for the location. Additionally, a **Maintain User Groups** option enables you to create user groups in mass for all the existing locations.

Several dependent processes must occur before Location Security is fully implemented:

1. When a Location is created, User Groups are created automatically in Oracle Hyperion Shared Services.
The user group contains the name of the location and additional prefix and suffix information based on the user preference. In addition, roles are provisioned for User Groups.
2. The administrator provisions the users to the User Groups.
3. When the user logs in, FDMEE determines the groups assigned to the user.
Based on the name of the group, FDMEE determines the accessible locations.
4. The POV region filters the locations based on the user access.



Note:

If the web services and batch scripts are used, then location security is still maintained and enforced.

To display the Location Security tab:

1. On the **Setup** tab, under **Configure**, select **Security Settings**.
2. Select the **Location Security** tab.

To add a user group for location security:

1. On the **Setup** tab, under **Configure**, select **Security Settings**.
2. Select the **Location Security** tab.
3. In the Location summary grid, click **Add**.

A *LOCATION* name row is added. When the group is saved, the Group name is in the form of Prefix_Location_Suffix, for example, *FDMEE_LOCATION_DATA*.

The prefix and suffix help identify groups in Common Shared Services (CSS).

4. In the **Security Setting Details** grid, enter a description of the user group in the **Description** field.

For example, enter: Group for Creating and Running Integration.

5. In the **Prefix** field, enter **FDMEE**.

When the group is saved, the prefix is prepended to the group name.

**Note:**

Underscore is not supported in the prefix or suffix for group names.

6. In the **Suffix** field, select the name of the function or rule that the user is provisioned to access.

**Note:**

Underscore is not supported in the prefix or suffix for group names.

For example, specify:

- Data Rule Metadata Rule (Run Integration role)
- HR Rule (HR Integration role)
- Create Integration
- Drill Through

When the group is saved, the suffix is appended to the group name.

7. Select the list of roles provisioned for the user group by selecting the appropriate roles:
 - Create Integration
 - Drill Through
 - Run Integration
 - HR Integration
 - Intermediate 2-9

The list of roles is described in [Table 1](#).

8. Click **Save**.
9. To create users groups in mass for the location, click **Maintain User Groups**.

To disable security by location:

1. On the **Setup** tab, under **Configure**, select **Security Settings**.
2. Select the **Location Security** tab.
3. Click **Disable Security by location**.

When security by location is disabled, this message is displayed: Security by Location is disabled. Would you like to enable the feature?

4. Click **Save**.

Setting up Source Systems

In some cases, you have multiple general ledger or human resource source systems. You can use Oracle Hyperion Financial Data Quality Management, Enterprise Edition to extract data and metadata from any instance.

For information on the source systems that FDMEE supports, see the *Oracle Hyperion Enterprise Performance Management System Certification Matrix*.

1. Register a source system. See [Registering Enterprise Resource Planning \(ERP\) Source Systems](#).
2. Edit source system settings as necessary. See [Editing Registered Source System Details](#).

For information on removing a registered source system, see [Deleting Registered Source Systems](#).

**Note:**

For information on viewing FDMEE processes or jobs, see [Viewing Process Details](#).

Registering Enterprise Resource Planning (ERP) Source Systems

The source system page displays all registered source systems in a table in the Summary pane. By default, the following columns are displayed:

- Name—Name of the source system
- Type—Type of source system
- Description—The description that you entered when the source system was registered.
- Drill URL—The drill URL selected when the source system was registered.

To add a source system:

1. On the **Setup** tab, under **Register**, select **Source System**.
2. In **Source System**, click **Add**.
3. Enter the source system details:
 - a. In **Source System Name**, enter the source system name.
 - b. In **Source System Description**, enter a description of the source system.
 - c. In **Source System Type**, select the source system type.

Available source systems:

- E-Business Suite Release 11i
- PeopleSoft Financials Release 9
- Peoplesoft HCM Release 9
- Fusion Applications
- SAP ERP Financial
- SAP BW (Business Warehouse)
- JD Edwards Enterprise One
- File
- Others

d. Enter the **Drill URL**.

The Drill-Through URL identifies the URL to use for drilling through. For example, you might specify `http://machinename.us.oracle.com:6362`

The URL is used to launch Fusion, E-Business Suite, or PeopleSoft.

You can drill through to any location as long as a URL is available, or as long as the user can specify a JavaScript that provides the necessary drill-through options. Drill through is available for predefined adapters, files, and open interface sources.

Additionally you can drill through to the Oracle General Ledger or PeopleSoft Enterprise Financial Management from an Enterprise Performance Management (EPM) system application that displays data loaded from the source system. When you click a hyperlink, you can navigate to the Oracle General Ledger Balances page or PeopleSoft Enterprise Financial Management Inquiry page.

For JD Edward source systems, you can drill through to the JD Edwards balances page.



Note:

Drill through is not supported for Oracle Hyperion Financial Management journals and intercompany transactions.

e. In **ODI Context Code**, enter the context code.

The ODI context code refers to the context defined in Oracle Data Integrator. A context groups the source and target connection information.

The default context code is **GLOBAL**.

4. **Optional:** If you use PeopleSoft's Commitment Control functionality, select **Enable Commitment Control**.

See [PeopleSoft Commitment Control](#).

5. Click **Save**.

After you add a source system, you can select the source system in the table, and the details are displayed in the lower pane.

After you register a source system, you must initialize the source system. Initializing the source system fetches all metadata needed in Oracle Hyperion Financial Data Quality Management, Enterprise Edition, such as ledgers, chart of accounts, and so on. It is also necessary to initialize the source system when there are new additions, such as chart of accounts, segments/chartfields, ledgers, and responsibilities in the source system.

6. To initialize a source system, click **Initialize**.



Note:

Depending on the size of the source system, initializing might take several minutes.

Registering File-Based Source Systems

The source system page displays all registered source systems in a table in the Summary pane. By default, the following columns are displayed:

- Name—Name of the source system
- Type—A file-based source system is the only supported source system.
- Description—The description that you entered when you registered the source system.
- Drill URL—The drill URL you entered when you registered the source system.

To add a file-based source system:

1. On the **Setup** tab, under **Register**, select **Source System**.
2. Click **Save**.

After you add a source system, you can select the source system in the table, and the details are displayed in the lower pane.

Deleting Registered Source Systems

You can delete registered source systems if you do not plan to use the source system with Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

Caution:

Use caution when deleting registered source systems! Part of the procedure for deleting a source system is to delete the target application. When you delete the target application, other artifacts are deleted. When you delete a registered source system, the source system is removed from the **Source System** screen, and all import formats, locations, metadata rules, and data rules associated with the source system are removed.

To remove a registered source system:

1. On the **Setup** tab, under **Register**, select **Target Application**.
Use the Target Application page to remove all target applications that have rules or mappings to the source system.
2. On the **Setup** tab, under **Register**, select **Source System**.
3. In **Source System**, select the source system to remove, and then click **Delete**.

Tip:

To undo a deletion, click **Cancel**.

4. Click **OK**.

Editing Registered Source System Details

Sometimes, source system details change. You can edit the source system details as needed. Keep in mind that after you add a source system type, you should not modify it.

To edit registered source system settings:

1. On the **Setup** tab, under **Register**, select **Source System**.
2. Select the source system.
3. Edit the source system details or ODI details as necessary.
4. Click **Save**.

If you make metadata changes in the source system, (for example, you add a new segment, chartfield values, or hierarchies), you must initialize the source system.

5. Click **Initialize**.

Adding File-Based Data Load Definitions

Source systems with the type "file" are used in import formats to load data from fixed and delimited files.

Oracle Hyperion Financial Data Quality Management, Enterprise Edition creates a file-based data load system automatically. If you need to create an alternate file-based data load source system, follow the procedure below.

To use file-based import formats, you must define a file-based data load.

To add a file-based data load definition:

1. On the **Setup** tab, under **Register**, select **Source System**.
2. In **Source System**, click **Add**.
3. Enter the source system details:
 - a. In **Source System Name**, enter the file-based data load system name.
 - b. In **Source System Description**, enter a description.
 - c. In **Source System Type**, select **File**.
 - d. In **Drill Through URL**, specify the URL that identifies the URL to use for drilling through.
 - e. In **ODI Context Code**, enter the context code.

The ODI context code refers to the context defined in Oracle Data Integrator. A context groups the source and target connection information.

4. Click **Save**.

After you add a file-based data load system, select the source system in the table. The details of the system are displayed in the lower pane.

Working with Source Adapters

A source adapter is an integration framework in Oracle Hyperion Financial Data Quality Management, Enterprise Edition that enables you to extract data from source

system in a flexible and customizable manner. A source adapter consists of two components:

- Oracle Data Integrator (ODI) Project—Contains the code to extract and load the data from the source system to the FDMEE staging tables.
- Adapter Definition XML—Contains the definition of the integration. It contains three components: source columns, parameters and the drill URL.

Three pre-packaged integrations are delivered using this framework:

- SAP ERP Financials
- Open interface to load from any source system
- JD Edwards

Using a Prepackaged Integration for SAP

For prepackaged integrations for SAP, review the instructions in the readme to download the necessary information from Oracle's integration partner. Oracle delivers the Oracle Data Integrator (ODI) Project and an Adapter Definition XML file. For the SAP integrations, the integration partner delivers the ODI Project and Adapter XML.

To use the prepackaged integration:

1. Use the ODI Console to import the **Project & Model definitions** into the Work Repository setup for Oracle Hyperion Financial Data Quality Management, Enterprise Edition.
2. Copy the **Adapter Definition XML** to a location under the directory that has been set up as the Application Root directory in System Settings.
3. Use the Source Adapter screen to import the Adapter XML file.

To import a file:

- a. On the **Setup** tab, under **Configure**, select **Source Adapter**.
- b. In the **Source Adapter**, select **Import**.
- c. In **Select file to import**, navigate to the file to import, and then click **OK**.

The file name depends on the adapter desired. For example, select:

- SAP_GLNew_Adapter.xml
- SAP_GLClassic_Adapter.xml
- SAP_PC_Adapter.xml
- SAP_CC_Adapter.xml
- SAP_AP_Adapter.xml
- SAP_AR_Adapter.xml

- d. **Optional:** To browse for a file, select **Upload**.
 - e. In **Select a file to upload**, click **Browse** to navigate to the file to import, and then click **OK**.
4. Create an import format of new type **Source Adapter** that defines the mapping between the source columns (identified in the source adapter) and the target application dimensions.
 5. Define the **Locations** and **Data Rule**.

When you execute the data rule, the new ODI Scenario associated with the import format is used to extract the data from the source and stage it in the FDMEE staging table (TDATESEG).

Adding Source Adapter Definitions

Use the Source Adapter screen to review prepackaged adapters or to customize prepackaged adapters. Do not directly modify the prepackaged adapters. As Oracle ships updates to the prepackaged adapter, the changes you make are overwritten. Always make a copy of the prepackaged adapter and make the necessary changes. You can also use this screen to create new custom adapters for custom load integrations.

The Source Adapter screen includes three sections:

- Source adapter information summary. See [Defining Source Adapter General Information](#).
- Source adapter detail. See [Defining Source Adapter Detail](#).
- Three sub-tabs from which you select: source columns information, parameters to pass to the ODI scenario, and URLs for drill through.

See:

- [Selecting Source Columns](#)
- [Defining Parameters](#)
- [Defining URLs for Drill Through](#)

Defining Source Adapter General Information

Use the Source Adapter Summary section to view, define, delete, export, import, and copy summary information about a source adapter.

Adding a Source Adapter

To add a source adapter:

1. On the **Setup** tab, under **Register**, select **Source Adapter**.
2. In **Source Adapter**, from the **Source Adapter** summary task bar, click **Add**.
Blank entry rows are added to the Source Adapter summary section.
Complete the following steps in the Source Adapter details section.
3. In **Adapter Key**, enter a user defined identifier for the adapter.
For example, for an Open Interface source adapter, enter OPENINT. You cannot modify the value in this field after it has been created.
4. In **Adapter Name**, enter a user defined name for the adapter.
For example, enter **Open Interface Adapter**.
5. In **ODI Object Name**, enter the name of the package or scenario.
6. In **ODI Object Type**, select **Package** or **Scenario**.
 - **Package**—A Package is a sequence of steps organized into an execution diagram. Packages are the main objects used to generate scenarios for

production. When you select Package as the ODI object type, then the source columns are shown, and you can specify maps in the import format in which the scenario is generated.

- Scenario—ODI compiled code that must match exactly as specified in the ODI.
7. In **ODI Object Project code**, enter the template ODI project code.
 8. In **Source System Type**, select the source application type.
 9. Click **Save**.

Deleting a Source Adapter

Deleting a source adapter removes the parameters, drill-down URLs and source columns associated with the source adapter.

To delete a source adapter:

1. On the **Setup** tab, under **Register**, select **Source Adapter**.
2. In the **Source Adapter** summary grid, select the source adapter and click **Delete**.
The message is displayed: This also deletes associated Parameters, Drill Down URLs and Source Columns. Delete Source Adapter?
3. Click **OK**.

Exporting a Source Adapter

Use the Export feature to migrate an adapter definition from one environment to another in the XML file format.

To export a file:

1. On the **Setup** tab, under **Register**, select **Source Adapter**.
2. In **Source Adapter**, on the **Source Adapter** summary task bar, in **Export**, select **Export Current** or **Export All**.
 - Export Current—Exports a selected source adapter from the Source Adapter Summary list.
 - Export All—Exports all source adapters from the Source Adapter summary list.
3. To export a selected file, in **Specify file location**, navigate to the file to export, and then click **OK**.
You can enter the full path name and file name in the **File Name** field.
4. **Optional:** From **Open File**, save the file to a local directory on the desktop.
5. **Optional:** To browse for a file to export, select **Upload**, and then browse to the file to export. When the File Upload screen is displayed, navigate to the file to upload, and then click **Open**.
6. Click **OK**.

Importing a Source Adapter

To import a source adapter:

1. On the **Setup** tab, under **Register**, select **Source Adapter**.

2. In **Source Adapter**, on the **Source Adapter** summary task bar, select **Import**.
3. In **Specify file to import**, navigate to the file, and then click **OK**.
4. In **Open File**, open the XML file or save it to a directory of your choice.
5. **Optional:** To browse to a file, select **Upload**. When the **File Upload** screen is displayed, navigate to the file, and then click **Open**.
6. Click **OK**.

The source adapter is added to the summary list.

Copying a Source Adapter Key

You can copy a source adapter key and its associated detail to a new adapter key.

To copy a source adapter key:

1. On the **Setup** tab, under **Register**, select **Source Adapter**.
2. In **Source Adapter**, on the **Source Adapter** summary task bar, select **Copy**.
3. In **Copy**, enter the name of the new adapter key, and then click **OK**.

Using the SAP BW Source Adapter

You can use an SAP Business Warehouse (BW) source adapter to extract metadata and data from an SAP BW source system. The SAP BW integration with Oracle Hyperion Financial Data Quality Management, Enterprise Edition differs from the standard and prepackaged integration frameworks. SAP (BW) comes with many predefined InfoCubes, or you can define your own InfoCubes.

Dimensions and hierarchies can be extracted from the SAP BW cube. After selecting the Infocube from the Source Adapter screen, metadata rules can be created to bring over dimensions and hierarchies.

To add an SAP BW source adapter:

1. On the **Setup** tab, under **Register**, select **Source Adapter**.
2. In **Source Adapter**, on the **Source Adapter** summary task bar, click **Add**.
Blank entry rows are added to the Source Adapter summary section.
3. In the **Source Adapter** details grid, in **Adapter Key**, enter a user defined identifier for the adapter.
For example, for an SAP BW source adapter, enter NK_SAP. The value in this field cannot be modified after it is created.
4. In **Adapter Name**, enter a user defined name for the SAP BW adapter.
For example, enter NK_SAP.
5. In **ODI Package Name**, enter the name of the package.
For example, enter NK_SAP.
6. In **ODI Project Code**, enter the template ODI project code name.
The ODI Object Project code identifies an object within the project.
7. In **Source System Type** select the **SAP BW** source system.
8. In **Cube Name**, enter the predefined InfoCube name.

9. Click **Import Cube Definition**.

The SAP BW cube information is imported into FDMEE, which creates the source columns in the Source Column tab.

10. In the **Source Column** grid, map each SAP BW cube column in the **Column Name** field to the type of data column in **Classification**.

For example, for the IOPK_0FIGL_C101_0C0_AREA_PK_C0_AREA column, select the Fact classification.

Classification types:

- Fact
- Year
- Period
- Period Number

11. In **Display Name**, enter the name to display for the column.

The name is shown on the Import Format screen and Data Load Workbench.

12. Select the **Parameters** tab, and specify any filter parameters.

See [Defining Parameters](#).

13. In the **Source Adapter** summary grid, click **Generate Template Package**.

The template package is used to generate the final scenario based on the import format mappings. If any customizations are required, you can customize the auto generated template package.

14. Click **Save**.

15. Define the import format and generate the process.

Defining Source Adapter Detail

When you have associated the source adapter with a source system type, the next steps include:

- Selecting Source Columns. See [Selecting Source Columns](#).
- Defining Parameters. See [Defining Parameters](#).
- Defining URLs for drill through. See [Defining URLs for Drill Through](#).

Selecting Source Columns

Use source columns to specify the columns that are available for mapping in the import format. There is no validation of the table or column names in this section—the source columns are registered so that they are available in import format for mapping.

Adding Source Columns

The source columns added in this section must match the columns specified in the data model for the source table in the Oracle Data Integrator.

To add a source column:

1. On the **Setup** tab, under **Register**, select **Source Adapter**.

2. In **Source Adapter**, select the **Source Column** tab.
3. From the **Source Column** task bar, select **Add**.
Entry fields are displayed for the Column Name, Table Name, and Display Name columns.
4. In **Column Name**, enter the source column name.
For example, enter SEGMENT1. The source column name must have a corresponding column name in the ODI model.
5. In **Table Name**, enter the table name.
For example, enter LOGICAL_BALANCES. The table name must have a corresponding table name in the ODI model.
6. In **Display Name**, enter the display name used for the import format.

Defining Parameters

Use the Parameters tab to specify the list of parameters for an ODI Scenario. The parameter definition includes a non-translated parameter name and a translated prompt. The prompt is used for display on the data rule page.

This section includes information on:

- Adding Parameters. See [Adding Parameters](#).
- Deleting Parameters. See [Deleting Parameters](#).

Adding Parameters

When you add a parameter for the source adapter, a definition is created, which includes a non-translated parameter name and a translated prompt.

To add a source column:

1. On the **Setup** tab, under **Register**, select **Source Adapter**.
2. In **Source Adapter**, select the **Parameters** tab.
3. From the **Parameters** task bar, select **Add**.
Entry fields are displayed for the Parameter Name, Parameter Data Type, and Parameter Prompt columns.
4. In **Parameter Name**, enter the parameter name.
For example, enter p_actual_flag. This is the name of the parameter in the ODI project.
5. In **Parameter Data Type**, select the data type of the parameter:
 - Char
 - Number
 - Date
6. In **Parameter Prompt**, enter the parameter prompt.
For example, enter Balance Type in this row.

 Note:

When a new parameter is added and you are in a multi-language environment, then set the browser locale to the appropriate language and edit the prompts as needed in that language.

Deleting Parameters

To delete a parameter:

1. On the **Setup** tab, under **Register**, select **Source Adapter**.
2. In **Source Adapter**, select the **Parameters** tab.
3. On the **Parameters** task bar, select the parameter, and then click **Delete**.
4. Click **OK**.

Defining URLs for Drill Through

Oracle Hyperion Financial Data Quality Management, Enterprise Edition provides a framework for using URLs for drill through. You click the hyperlink in the amount cell, and the source system is launched in a new EPM Workspace tab or a new window. Multiple drill through URLs are provided to drill to different pages in the source system based on the import format mapping.

See also [Drilling Through](#).

Adding Drill Through URLs

Use these steps to add the drill through URL in Oracle Hyperion Financial Data Quality Management, Enterprise Edition. There is a server component and a detailed component when defining the drill URL. The server component is added to the source system definition and the detailed component is added to the import format definition.

 Note:

For information on adding a drill through to file based source systems, see [Using Drilling Through](#).

To create a drill through URL, specify the URL name, URL, request method, and URL prompt.

To add a drill through URL:

1. On the **Setup** tab, under **Register**, select **Source Adapter**.
2. In **Source Adapter**, select the **Drill URL** tab.
3. On the **Drill URL** task bar, select **Add**.
Entry fields are displayed for the Drill URL Name, Drill URL, Request Method, and Drill URL Prompt columns.
4. In **Drill URL Name**, enter a user defined name for the drill through URL.

5. In **Drill URL**, enter the **URL** used for the drill through.

Enter the URL without the server and port information. The URL must contain the parameter name and column name from the TDATESEG table enclosed in the symbol \$.

For example, enter: LEDGER_ID=\$ATTR1\$&GL_PERIOD=\$ATTR2\$.

In the above example the value of ATTR1 is passed as a value for the LEDGER_ID parameter, and ATTR2 is passed as the value for the GL_PERIOD parameter. Parameters are separated by the "&" character.

To specify the request-response between a client and server for the drill URL format, enter either:

- GET—Form data is encoded into the URL. For example, specify: GET@http://www.oracle.com/. If no method is specified, then GET is the assumed request-response.
- POST—Form data is displayed in the message body. For example, specify: POST@http://www.oracle.com/.

**Note:**

A JavaScript specification can also be used for the drill-through. See [Drill Through using JavaScript](#).

6. In the **Drill URL Prompt**, enter a user-defined prompt for the drill-through prompt. For example, enter Default.

Drill Through using JavaScript

You can specify a JavaScript based drill through that enables you to drill through to a graphical user interface instead of an URL. To do this, you write a script in the Script Editor and store it in a custom script directory.

The JavaScript for the drill through is associated with the source adapter. You invoke JavaScript from the drill through link by specifying "#javascript" instead of "http URL" in the source system. You can also pass contextual parameters to a JavaScript function similar to the http drill URL.

To add a drill through using JavaScript:

1. On the **Setup** tab, under **Register**, select **Source System**.
See [Setting up Source Systems](#).
2. Select the source system file for the JavaScript based drill through.
3. In the detail section, from **Drill Through URL**, enter **#javascript**.
4. Click **Save**.
5. On the **Setup** tab, under **Scripts**, select **Script Editor**.
6. On the **Script Editor**, click **New**.
7. From the **Create Script Options** window, in **Script Type**, select **Drill**.

 Note:

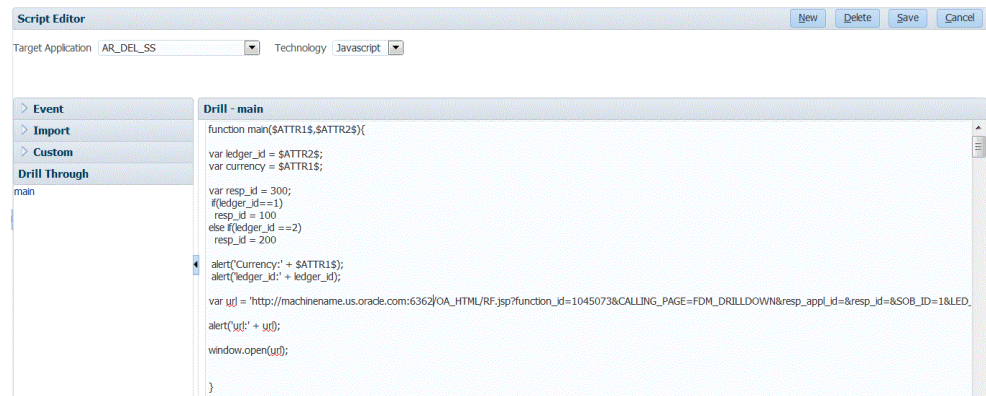
The Technology drop-down is JavaScript because drill through scripts can only be created based on JavaScript.

8. In **File Name**, enter a name for the script.
9. Click **OK**.
10. In **Target Application**, select the target application for this drill through script.
11. Write the code for the custom script in the Script Editor.

Ensure the name of the JavaScript function is same as the file name.

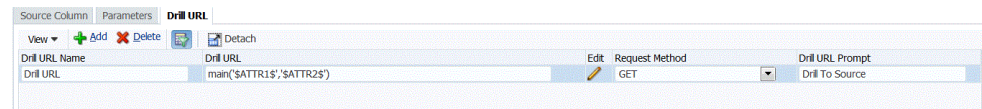
For example,

```
function main($ATTR1$, $ATTR2$){  
  
    var ledger_id = $ATTR2$;  
  
    var currency = $ATTR1$;  
  
    var resp_id = 300;  
  
    if(ledger_id==1)  
        resp_id = 100  
  
    else if(ledger_id ==2)  
        resp_id = 200  
  
    alert('Currency:' + $ATTR1$);  
  
    alert('ledger_id:' + ledger_id);  
  
    var url = 'http://machinename.us.oracle.com:6362/OA_HTML/RF.jsp?  
function_id=1045073&CALLING_PAGE=FDM_DRILLDOWN&resp_appl_id=&resp_id=&SOB_ID=1&LE  
D_ID='+ledger-id+'&fdm_per=Jan-03&fdmccid=13098&fdm_currency='+currency  
+'fdm_balance_type=A  
  
    alert('url:' + url);  
  
    window.open(url);  
  
}
```



12. Click **Save**.
13. On the **Setup** tab, under **Register**, select **Source Adapter**.
See [Working with Source Adapters](#).
14. In **Source Adapter**, select the **Drill URL** tab.
15. On the **Drill URL** task bar, select **Add**.

Entry fields are displayed for the Drill URL Name, Drill URL, Request Method, and Drill URL Prompt columns.



16. In **Drill URL Name**, enter a user defined name for the drill through URL.
17. In the **Drill URL**, enter the Function name of the drill through.
The Function name should be same as the file name.
For example, enter: `main($ATTR1$, $ATTR2$)`.
18. In **Drill URL Prompt**, enter a user-defined prompt for the drill-through prompt.
19. Click **Save**.

When you drill down from the workbench, a new window is launched with the specified URL.

Deleting Drill URLs

To delete a drill through URL:

1. On the **Setup** tab, under **Register**, select **Source Adapter**.
2. In **Source Adapter**, select the **Drill URL** tab.
3. From the **Drill URL** summary area, select the drill-through URL, and then click **Delete**.
4. In **Delete Source Adapter Drill URL**, click **OK**.

Registering Target Applications

Target applications enable Oracle Hyperion Financial Data Quality Management, Enterprise Edition to be used as a primary gateway to integrate data between different source systems and target applications. In this way, you can deploy local Oracle Enterprise Performance Management Cloud applications, service instance to service instance deployments (cloud to cloud applications), and custom applications into your existing EPM portfolio. The process to integrate source applications with target applications provides data visibility, integrity, and verification systems.

The following application types describe the types of target application that can be used:

- **local**—This application type refers to a local EPM application (on-premise deployment) in the current service.

You might use the integration to import data from existing on-premise ERP applications, or synchronize data between on-premise EPM applications.

For example, Oracle Hyperion Financial Management customers can add Oracle Hyperion Planning, data or a Planning customer can add more Planning applications. In addition, this integration enables you to write back from a cloud to an on-premise application or other external reporting applications.

- **Cloud**—This application type refers to a service instance that uses a remote service to integrate data. A service instance is a self-contained unit often containing the web server and the database application. In this case, connection information must be selected between the two service instances. Service instances available include:

This feature enables EPM customers to adapt cloud deployments into their existing EPM portfolio including

- Oracle Enterprise Planning and Budgeting Cloud
- Oracle Planning and Budgeting Cloud
- Oracle Financial Consolidation and Close Cloud
- Oracle Hyperion Profitability and Cost Management
- Oracle Tax Reporting Cloud

To register a target application:

1. Select the **Setup** tab, and then under **Register**, select **Target Application**.
2. In **Target Application**, in the summary grid, click **Add**, and then select the type of deployment.

Available options are **Cloud** (for a Cloud deployment) or **Local** (for an on-premise deployment).

For a Cloud deployment, go to step 3.

For a Local deployment, go to step 4.

3. To register a Cloud deployment, select **Cloud** and then complete the following steps on the EPM Cloud Credentials screen:
 - a. In **URL**, specify the service URL that you use to logon to your service.
 - b. In **User name**, specify the user name for the Cloud Service application.

- c. In **Password**, specify the password for the Cloud Service application.
- d. In **Domain**, specify the domain name associated with the Cloud Service Application.

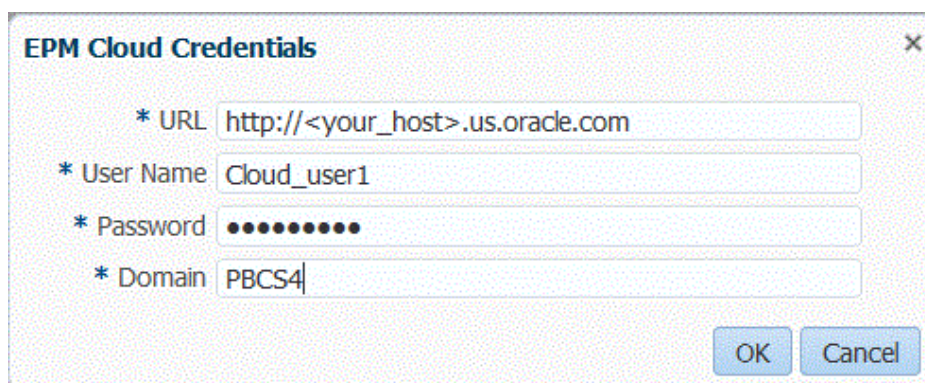
An identity domain controls the accounts of users who need access to service instances. It also controls the features that authorized users can access. A service instance belongs to an identity domain.

 Note:

Administrators can update the domain name that is presented to the user, but FDMEE requires the original domain name that was provided when the customer signed up for the service. Alias domain names cannot be used when setting up EPM Cloud connections from FDMEE.

- e. Click **OK**.

You can add or edit EPM Cloud credentials on the Application Options tab of the target application.



The image shows a dialog box titled "EPM Cloud Credentials" with a close button (X) in the top right corner. It contains four labeled input fields: "* URL" with the text "http://<your_host>.us.oracle.com", "* User Name" with the text "Cloud_user1", "* Password" with masked characters (dots), and "* Domain" with the text "PBCS4". At the bottom right, there are two buttons: "OK" and "Cancel".

- 4. From **Select Application**, and then from the **Type** drop-down, select the target application name and click **OK**.
- 5. To register a target application with the same name as an existing target application, in **Prefix**, specify a prefix to make the name unique.

The prefix name is joined to the existing target application name. For example, if you want to name a demo target application the same name as the existing "Vision" application, you might assign the **Demo** prefix to designate the target application with a unique name. In this case, the FDMEE joins the names to form the name **DemoVision**.

It is recommended when specifying the prefix, use between three to six characters.

- 6. In **Application Details**, enter the application name.
- 7. Click **OK**.
- 8. If using an Essbase database, select the **Essbase Database Name**.
- 9. Click **Refresh Members**.
- 10. Click **Save**.

11. Define the dimension details.

See [Defining Application Dimension Details](#).

 Note:

Do not modify dimensions in the EPMA application after you have registered the target application in FDMEE. Any action such as sharing or removing the dimension, and then re-adding it, alters the dimensions ID and creates a mismatch between EPMA and FDMEE.

Optional: If not all dimensions are displayed, click **Refresh Metadata**

12. Select the application options.

For Planning applications, see [Defining Application Options for Essbase and Planning](#).

For Oracle Hyperion Public Sector Planning and Budgeting, see [Defining Application Options for Public Sector Planning and Budgeting](#).

For Financial Management, see [#unique_124](#).

 Note:

No application options are available for the Account Reconciliation Manager.

Creating a Custom Target Application

A custom target application enables you to load data from an EPM supported source, and then extract the data into a flat file instead of loading it to Oracle Essbase, Oracle Hyperion Planning, and Oracle Hyperion Financial Management. You can define the custom target application with required dimensionality. Instead of exporting the data to a target application, Oracle Hyperion Financial Data Quality Management, Enterprise Edition generates a data file that can be loaded to an external system using a custom process.

When creating a custom target application, note the following:

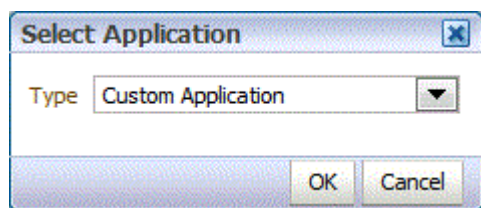
- Data is written to the file in the following predefined order: Account, Entity, UD1, UD2 ... UD20, AMOUNT.
- The sequence that is specified when you create the custom application definition is used to order the dimensions for mapping processing. Note the sequence in case you have scripts that have an order of processing dependency.
- Data Rule—The POV category is not validated.
- Data Load Mapping—Target values are not validated for custom applications.
- Data Load Execution—When the option to export the data file is enabled, FDMEE creates an output data file. The name of the data file is <Target App Name>_<Process ID>.dat, and it is written to the <APPL ROOT FOLDER>/outbox directory. You can access the data file from the Process Details page from the OUTPUT file column.

When the data load rule is executed, FDMEE exports the data.

If you want to create a custom file, then set the **Enable Export to File** option to **No** and then write a custom BefExport event script to create a data file. If you want to create a custom process to load target script automatically, then write a BefExport script.

To define a custom target application:

1. On the **Setup** tab, under **Register**, select **Target Application**.
2. In **Target Application**, in the **Target Application** summary grid, click **Add**.
3. Select **Local** target application.
4. From **Select Application**, select **Custom Application**, and then click **OK**.



5. In **Application Details**, enter the application name.
6. Select the **Dimension Details** tab.
7. Specify the **Dimension Name**.
8. Select the **Target Dimension Class** or click  to select the **Target Dimension Class** for each dimension that is not defined in the application.

The dimension class is a property that is defined by the dimension type. For example, if you have a Period dimension, the dimension class is also “Period.” For Essbase applications, you must specify the appropriate dimension class for Account, Scenario, and Period. For Oracle Hyperion Public Sector Planning and Budgeting applications, you must specify the dimension class for Employee, Position, Job Code, Budget Item, and Element.

9. In **Data Table Column Name**, specify the table column name of the column in the staging table (TDATESEG) where the dimension value is stored.

Click  to search and select a data table column name.

10. In **Sequence**, specify the order in which the maps are processed.
For example, when Account is set to **1**, Product is set to **2**, and Entity is set to **3**, then FDMEE first processes the mapping for Account dimension, followed by Product, and then by Entity.
11. In **Prefix Dimension for Duplicates**, enable or check (set to **Yes**) to prefix member names by the dimension name.

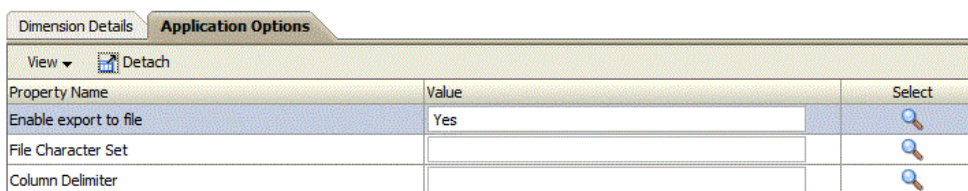
The member name that is loaded is in the format [Dimension Name]@[Dimension Member]. The prefixed dimension name is applied to all dimensions in the application when this option is enabled. You cannot select this option if there is a dimension in the target that has duplicate members. That is, only select this option when the duplicate members are across dimensions.

If the application supports duplicate members and Prefix Dimension for Duplicates is disabled or unchecked (set to **no**), then the user must specify the fully qualified member names. Refer to the Essbase documentation for the fully qualified member name format.

 Note:

Planning does not support duplicate members.

12. Click **Application Options**.



Property Name	Value	Select
Enable export to file	Yes	
File Character Set		
Column Delimiter		

13. In **Enable export to file**, select **Yes** to have FDMEE create an output data file for the custom target application.

A file is created in the outbox folder on the server with the following name format: <LOCATION>_<SEQUENCE>.dat. For example, when the location is named **Texas** and the next sequence is **16**, then the file name is **Texas_15.dat**. The file is created during the export step of the workflow process.”

When the **Enable export to file** option is set to **No**, then the Export to Target option is unavailable in the execution window.

14. In **File Character Set**, select the file character set.

The file character set determines the method for mapping bit combinations to characters for creating, storing, and displaying text. Each encoding has a name; for example, UTF-8. Within an encoding, each character maps to a specific bit combination; for example, in UTF-8, uppercase A maps to HEX41.

15. In **Column Delimiter**, select the character to use for delimiting columns in the output file.

Available column delimiters are:

- ,
- |
- !
- ;
- :

16. In **File Name to Download**, enter the file name to copy.

17. Click **Save**.

Adding Lookup Dimensions

Lookup dimensions can be created and assigned with data columns for target applications. Lookup dimensions are used for mapping and reference. They can only be used in Oracle Hyperion Financial Data Quality Management, Enterprise Edition and do not affect the dimensionality of a target application. They can also be used with member mapping functionality to cross-reference multiple source segments and chartfields and assign a target value.

To add a lookup dimension:

1. On the **Setup** tab, under **Register**, select **Target Application**.
2. In the **Target Application** summary grid, select a target application.
3. Select the **Dimension Details** tab.
4. Click **Add**.

Blank dimension name and data table column name entry fields are displayed.

5. In **Dimension Name**, enter the lookup dimension name.
6. In **Data Table Column Name**, select the data column from which to base the lookup dimension.



Note:

The data table column name value must be a user-defined dimension greater than the selected target dimension. For example if the application has four custom dimensions, select **UD5**.

7. Click **OK**.

The lookup dimension is added to the dimension detail list with the target dimension class name of "LOOKUP." To use the lookup dimension as a source dimension, make sure you map it in the import format.

Defining Application Dimension Details

The dimension details differ for each application type. For Oracle Hyperion Public Sector Planning and Budgeting and Oracle Essbase applications, you reclassify the dimensions and change the dimension class, as necessary.

To define dimension details:

1. On the **Setup** tab, under **Register**, select **Target Application**.
2. In the **Target Application** summary grid, select an Essbase or Oracle Hyperion Planning application.
3. Select the **Dimension Details** tab.
4. Select the **Target Dimension Class** or click  to select the **Target Dimension Class** for each dimension that is not defined in the application.

The dimension class is a property that is defined by the dimension type. For example, if you have a Period dimension, the dimension class is also "Period." For

Essbase applications, you must specify the appropriate dimension class for Account, Scenario, and Period. For Public Sector Planning and Budgeting applications, you must specify the dimension class for Employee, Position, Job Code, Budget Item, and Element.

5. **Optional:** Click **Refresh Metadata** to synchronize the application metadata from the target application.
6. In **Data Table Column Name**, specify the table column name of the column in the staging table (TDATESEG) where the dimension value is stored.

Click  to search and select a data table column name.

7. In **Sequence**, specify the order in which the maps are processed.
For example when Account is set to **1**, Product is set to **2**, and Entity is set to **3**, then Oracle Hyperion Financial Data Quality Management, Enterprise Edition first processes the mapping for Account dimension, followed by Product, and then by Entity.

8. Click **Save**.

The target application is ready for use with FDMEE.

**Tip:**

To edit the dimension details, select the target application, then edit the application or dimension details, as necessary. To filter applications on the Target Application page, ensure that the filter row is displaying above the column headers. (Click  to toggle the filter row.) Then, enter the text to filter.

Defining Application Options for Essbase and Planning

After defining the application details and dimension details, for Oracle Essbase and Oracle Hyperion Planning, define the application options.

**Note:**

The user attempting to load data to Planning must be provisioned with Essbase administrator or Planning administrator rights.

To define application options for Essbase or Planning applications:

1. On the **Setup** tab, under **Register**, select **Target Application**.
2. In the **Target Application** summary grid, select a Planning target application or Essbase target application.
3. After defining the application details and dimension details in **Application Detail**, select the **Application Options** tab.
4. Complete the application options as needed.
5. Click **Save**.

Table 1-12 Essbase and Planning Application Options and Descriptions

Option	Description
Purge Data File	When a file based data load to Essbase is successful, specify whether to delete the data file from the application outbox directory. Select Yes to delete the file, or No to retain the file.
Prefix Dimension for Duplicate Members	When this option is set to Yes, then member names are prefixed by the Dimension Name. The member name that is loaded is in the format [Dimension Name]@[Dimension Member]. The prefixed dimension name is applied to all dimensions in the application when this option is enabled. You cannot select this option if there is a dimension in the target that has duplicate members. That is, only select this option when the duplicate members cross dimensions. If the application supports Duplicate Members and the Prefix is set to No, then the user must specify the fully qualified member names. Refer to Essbase documentation for the fully qualified member name format.
	 Note: Planning does not support duplicate members.
Global User for Application Access	Option to override the Single Sign-On logon to the Essbase and the Planning applications. When a user name is specified for this option, this user name is used to access Essbase/Planning applications instead of the Oracle Hyperion Financial Data Quality Management, Enterprise Edition sign-on user name. Specify the user name of a user who has administrator access to the Planning application, and/or Application/Database Manager access to the Essbase applications.
Batch Size	Specify the batch size used to write data to file. The default size is 10,000.

Table 1-12 (Cont.) Essbase and Planning Application Options and Descriptions

Option	Description
Load Method	Specify the method for loading data from the TDATESEG staging table to Essbase. Available methods are: • File—Data is written to a data file in the outbox directory of the application (defined in System Settings). The file name is in the format <APPLICATION NAME>_<PROCESS_ID>.dat. It is then loaded to Essbase. The file load method creates an application file type of ESSFILE. • SQL—Uses SQL Method to load data. The SQL load method is the default. The SQL method creates an application file type of ESSSQL.
Check Entity Calculation Method	Specify the calculation method for check entities. Available methods are: • dynamic—Check entity data is calculated based on the data at retrieval time. If set to dynamic, then the default Essbase calculation is executed. • calculation script—Check entity data is calculated based on a predefined calculation script.
Source Language for Member Description	Select the default language for the member descriptions. To understand how languages are processed, see How Languages Are Processed.

Table 1-12 (Cont.) Essbase and Planning Application Options and Descriptions

Option	Description
Drill Region	Select Yes to create a drill region. A drillable region is created to use the drill through feature. Note: FDMEEE does not support drilling through to human resource data. When loading data from FDMEEE, the drill region is loaded to Planning data. FDMEEE creates drill region by scenarios. For any cube (Planning plan types or Planning databases, the name of the drill region is FDMEEE_<name of the scenario member>). When creating the drill region, FDMEEE checks if a dimension is enabled for the drill. Members of enabled dimensions selected in data loads, are included in the drill region filter. If no dimensions are enabled, the following dimensions are enabled by default: Scenario, Version, Year, and Period. You can enable additional dimensions, and the subsequent data load considers members of newly enabled dimensions. If you disable any dimensions, which were previously included in a drill region used for drill creation, members of such dimensions are not deleted during the subsequent data loads. If needed, you can remove obsolete members manually.
Date Format	Use the date format based on the locale settings for your locale. For example, in the United States, enter the date using the format MM/DD/YY format.
Data Dimension for Auto-Increment Line Item	Select the data dimension that matches the data dimension you specified in Planning. Used for loading incremental data using a LINEITEM flag. See Loading Incremental Data using the LINEITEM Flag to an EPM Application.
Driver Dimension for Auto-Increment Line Item	Select the driver dimension that matches the driver dimension you specified in Planning. Used for loading incremental data using a LINEITEM flag. See Loading Incremental Data using the LINEITEM Flag to an EPM Application.

Table 1-12 (Cont.) Essbase and Planning Application Options and Descriptions

Option	Description
Member name may contain comma	To export a dimension member name containing a comma to Planning, select Yes . Otherwise, select No .

Working with Duplicate Members in Essbase

The table below provides details how duplicate members are referenced in an Oracle Essbase application

Table 1-13 How Duplicate members are referenced in an Essbase application

Scenario	Qualified Name Syntax	Example
Duplicate member names exist at generation 2	[DimensionMember]. [DuplicateMember]	[Year].[Jan]
Duplicate member names exist in an outline but are unique within a dimension	[DimensionMember]@[DuplicateMember]	[Year]@[Jan]
Duplicate member names have a unique parent	[ParentMember].[DuplicateMember]	[East].[New York]
Duplicate member names exist at generation 3	[DimensionMember]. [ParentMember].[DuplicateMember]	[Products].[Personal Electronics]. [Televisions]
Duplicate member names exist at a named generation or level, and the member is unique at its generation or level	DimensionMember@[GenLevelName]@ [DuplicateMember]	[2006]@[Gen1][Jan]
In some scenarios, the differentiating ancestor method is used as a shortcut.	DifferentiatingAncestor]. [Ancestors...].[DuplicateMember]	[2006].[Qtr1].[Jan]

Using Calculation Scripts

When loading data to Oracle Essbase or Oracle Hyperion Planning, Oracle Hyperion Financial Data Quality Management, Enterprise Edition enables you to run custom calculation scripts. A calculation script is a series of calculation commands, equations, and formulas that enable you to define calculations other than those defined by the database outline. Calculation scripts are defined in Essbase and Planning.

Custom calculation scripts are supported in the target application and the check entity group. In the target application:

- parameterize the script so that you can specify a parameter for each script.
- assign scripts to run before or after the data load.
- assign a scope of applicability to scripts. The scope can be at the application, category, location, and data rule level. If scripts are assigned at the lowest level, then they take precedence over script at higher level. The data Rule is the lowest level and application is the highest level. Only scripts at one level are executed.

To add a calculation script:

1. On the **Setup** tab, under **Register**, select **Target Application**.

2. From the **Calculation Script** tab, select a calculation script.

3. Click  to display the Parameter window.

4. For Planning applications only, select the **Plan Type** to retrieve the parameters defined in the calculation script.

5. Click the **Add** to add a script parameter.

6. Select the parameter that has been defined for the script.

You can also click **Browse**, select the parameter, and click **OK**. You can also manually type in the parameter name.

7. In **Script Value** select a predefined parameter value, or select **Custom** to specify your own script value.

8. **Optional:** If you selected **Custom** in the **Script Value** field, enter the value in **Custom**.

For example, enter **0001**.

9. Click **OK**.

10. In **Script Name**, specify the name of the script.

11. Click  to add the parameters for the calculation script.

See [Using Calculation Scripts](#).

12. In **Script Scope**, select the scope of applicability,

Note that scripts assigned at the lowest level take precedence over scripts at higher level.

Available scopes in order of highest scope to lowest are:

- Application (default scope)
- Category
- Location
- Data Rule

13. In **Scope Entity**, select the specific value associated with the script scope.

For example, if the script scope is "Location," select the location associated with the application.

The Scope Entity is disabled for the Application Script Scope.

14. In **Event**, select the event that executes the calculation script.

Available events:

- Before Data Load
- After Data Load
- Before Check
- After Check

15. In **Sequence**, specify the order in which the script is executed.

Since multiple scripts can be executed for an event, this sequence value provides the numeric order in which each script is executed. You can enter any number, but the number has to be unique within an event.

16. Click **Save**.

Adding Calculation Script Parameters

You parameterize calculation scripts by identifying and defining the list of values in the scripts. Parameter values can be predefined for a list of values, for example, the POV Period, POV Location, POV Category, Check Group Entity, and all data load entities.

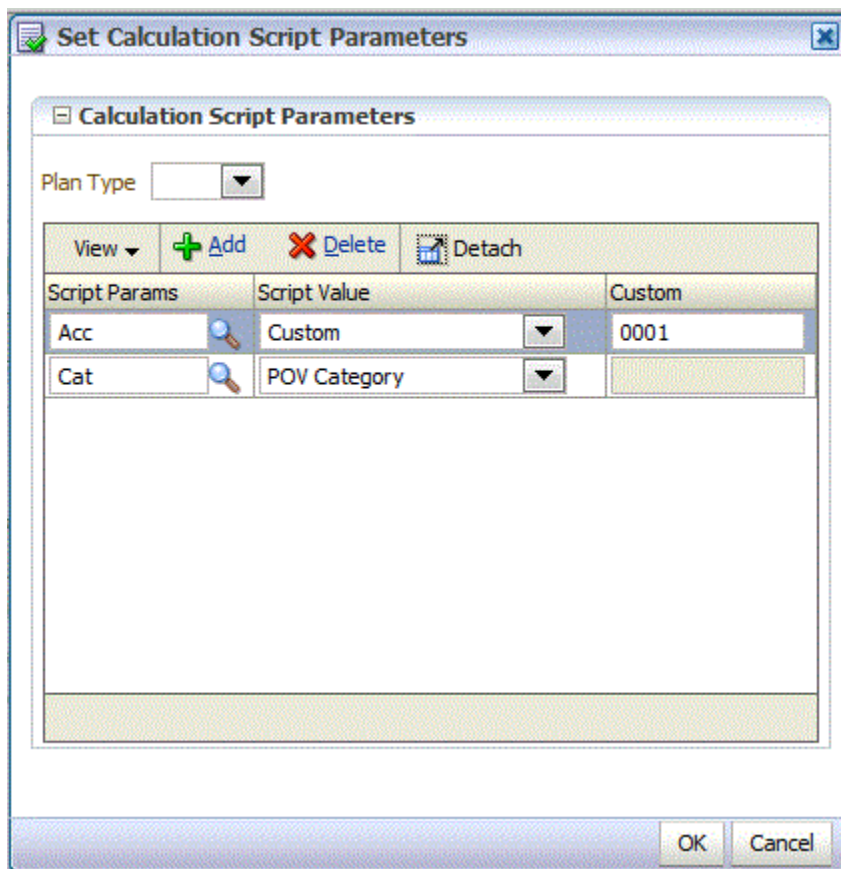
You can also specify custom parameters, in which case, you can enter any Oracle Essbase filter syntax.

Additionally, you can reference values stored in the Integration Option 1-4 fields in the Location table to drive calculation logic. Location Option 1, Location Option 2, Location Option 3, and Location Option 4 values can be referenced. In this case, the exact string that is passed as a parameter has to be stored in the Integration Option field of the Location table.

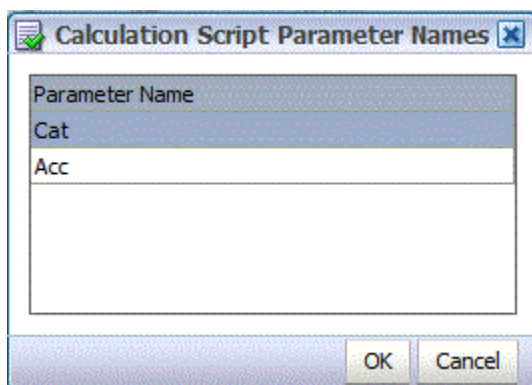
Similarly, you can reference four option fields in the Data Rule for parameters. These fields are Data Rule Option 1, Data Rule Option 2, Data Rule Option 3, and Data Rule Option 4. The column added to the Data Rule is called RULE_OPTION1, RULE_OPTION2, RULE_OPTION3 and RULE_OPTION4.

To add calculation script parameters:

1. From the **Calculation Script** tab, add or select a calculation script.
2. Click  to add the parameters for the calculation script.
3. For **Planning applications only**: select the **Plan Type** for Accounting and Entity dimensions.



4. In **Script Parameters**, select the parameter that has been defined for the script. You can click the **Browse** icon, select the parameter, and click **OK**.



5. In **Script Value**, select the value to which to apply the calculation script and click **OK**.

To add a custom script parameter:

1. From the **Calculation Script** tab, add or select a calculation script.
2. Click  to add the parameters for the calculation script.

3. In **Script Parameters**, select the parameter that has been defined for the script.
You can click the **Browse** icon, select the parameter, and click **OK**.
4. In **Script Value**, select **Custom** and click **OK**.
5. In **Custom**, enter the filter syntax for the custom value.
For example, enter **0001**.

Defining Application Options for Profitability and Cost Management (HPCM)

After defining the application details and dimension details, define the application options.

Oracle Hyperion Profitability and Cost Management uses EPMA for application definition and metadata management. It uses Oracle Essbase for data storage and calculation.

Profitability and Cost Management support only and EPMA deployment, and for this reason, available Profitability and Cost Management applications must be registered from the EPMA Application Library.

To define options for an Profitability and Cost Management application:

1. On the **Setup** tab, under **Register**, select **Target Application**.
2. In the **Target Application** summary grid, click **Add**.
3. In **Select Application**, in **Type**, select **Profitability**.
4. From **Name**, select the Profitability and Cost Management application and click **OK**.
5. After defining the application details and dimension details in **Application Detail**, select the **Application Options** tab.
6. Complete the application options as needed.
7. Click **Save**.

Table 1-14 Profitability and Cost Management Application Options and Descriptions

Property	Value
Drill Region	Select Yes to create a drill region. A drillable region is created to use the drill through feature for Profitability and Cost Management data.  Note: Oracle Hyperion Financial Data Quality Management, Enterprise Edition does not support drilling through to human resource data.
	The drill region URL allows Essbase, Oracle Smart View for Office, and Oracle Hyperion Financial Reporting to drill to the proper landing page. When loading data from FDMEE, the drill region is loaded to Oracle Hyperion Planning data. A drill region includes the Entity/Account/Scenario/Year/Period for Planning, a URL to get back to FDMEE, and a region name. For Essbase, you select the dimension to use for the drill region. Select Yes to enable or No to disable.
Source Language for Member Description	Select the default language for the member descriptions. To understand how languages are processed, see How Languages Are Processed.
Batch Size	Specify the batch size used to write data to file. The default size is 10,000.

Table 1-14 (Cont.) Profitability and Cost Management Application Options and Descriptions

Property	Value
Check Entity Calculation Method	Specify the calculation method for check entities. Available methods are: - dynamic—Check entity data is calculated based on the data at retrieval time. "dynamic" is the default check entity calculation method. - calculation script—Check entity data is calculated based on a predefined calculation script. If the calculation method is set to "dynamic", the default calculation is performed during Essbase consolidation. If the method is set to "calculation script", then the script name given on check entity screen is used to perform the consolidation in target system.
Prefix Dimension for Duplicate Members	When set to Yes member names are prefixed by the Dimension Name. The member name that is loaded is in the format [Dimension Name]@[Dimension Member]. The prefixed dimension name is applied to all dimensions in the application when this option is enabled. You cannot select this option if there is a dimension in the target that has duplicate members. That is, only select this option when the duplicate members cross dimensions. If the application supports Duplicate Members and Prefix is set to No, then the user is responsible to specify the fully qualified member names. Refer to Essbase Documentation for fully qualified member name format.

**Note:**

Planning does not support duplicate members.

Table 1-14 (Cont.) Profitability and Cost Management Application Options and Descriptions

Property	Value
Load Method	Specify the method for loading data from the TDATESEG staging table to Essbase. Available methods include: - File—Data is written to a data file in the outbox directory of the application (defined in System Settings). The file name is in the format <APPLICATION NAME>_<PROCESS_ID>.dat. It is then loaded to Essbase. The file load method creates an application file type of ESSFILE. - SQL—Uses SQL Method to load data. The SQL load method is the default. The SQL method creates an application file type of ESSSQL.
Purge Data File	When a file-based data load to Essbase is successful, specify whether to delete the data file from the application outbox directory. Select Yes to delete the file, or No to retain the file.

Defining Application Options for Public Sector Planning and Budgeting

After defining the application details and dimension details, for Oracle Hyperion Public Sector Planning and Budgeting applications, you define the application options. In Oracle Hyperion Financial Data Quality Management, Enterprise Edition, you must specify a parent member to populate all dimensions applicable for your model. You can optionally enter a parent member for earnings and benefits.

To define application options for Public Sector Planning and Budgeting Applications

1. On the **Setup** tab, under **Register**, select **Target Application**.
 2. In the **Target Application** summary grid, select a Public Sector Planning and Budgeting application.
 3. After defining the application details and dimension details in **Application Detail**, select the **Application Options** tab.
 4. Complete the application options as needed.
- Public Sector Planning and Budgeting application options are described in [Table 1](#).
5. Click **Save**.

The target Public Sector Planning and Budgeting application is ready for use with FDMEE.

Table 1-15 Public Sector Planning and Budgeting Application Options and Descriptions

Option	Description
Budget Model Type	Enter the Public Sector Planning and Budgeting model. (For Employee only, enter EMPLOYEE. For Position only, enter POSITION, or for both Employee and Position, enter Total Existing Positions).
Parent Member for all Positions	Enter the parent member for all positions imported from your human resource source system. For example, enter Total Existing Positions.
Parent Member for all Entities	Enter the parent member for all departments imported from your human resource source system. For example, enter Total Entity.
Parent Member for all Employees	Enter the parent member for all employees imported from your human resource. For example, enter Existing Employees.
Parent Member for Salary Elements	Enter the parent member for salary grades imported from your human resource source system. For example, enter Salary Grades.
Parent Member for Earnings Elements	Enter the parent member for earnings codes imported from your human resource source system. For example, enter Additional Earnings. Specifying parent members for earnings and benefit elements is optional. If you do not specify a parent member, the corresponding elements are not loaded into the application when you run the human resource data load rule.
Parent Member for all Job	Enter the parent member for all job codes imported from your human resource source system. (The parent member for job is applicable only to Employee Only models.)
Parent Member for Benefit Elements	Enter the parent member for benefit codes imported from your human resource source system. For example, enter Benefit Defaults.
Source Language for Member Description	Select the default language for the member descriptions. To understand how languages are processed, see How Languages Are Processed .
Drill Region	FDMEE does not support drilling through to human resource data. Select No for this option.

Defining Application Options for Financial Management

After defining the application details and dimension details, for Oracle Hyperion Financial Management application, you define the application options.

To define Financial Management application options:

1. On the **Setup** tab, under **Register**, select **Target Application**.
2. In the **Target Application** summary grid, select a Financial Management target application.
3. After defining the application details in **Application Detail**, select the **Application Options** tab.
4. Complete the application options as needed.
Financial Management application options are described below.
5. Click **Save**.

Table 1-16 Financial Management Application Options and Descriptions

Option	Description
Check Intersection	Enables the checking of Financial Management data intersections (account, entity, and so on) during the Validation step in the Data Load Workbench. The default setting is Yes .
Load Line Item Detail	Enables loading of line-item detail to Financial Management. Select Yes to enable or No to disable.
Line Item Detail Load Type	Specify whether to load line item detail or summary data to cells. The default is Load Detail , which displays details for the selected cell. (Specifies whether an account can have line items.) If this setting is Load Summarized , cells show summarized information. • Load Summarized • Load Detail
Enable Data Load	Enables the data load process. Select Yes to enable or No to disable.
Load Process	Select the process for loading data. Select Scan to scan the file for invalid records before loading it to the application. Select Load to load the file only. • Scan—Validates data and lists invalid records into the Log. When this option is selected, data is not loaded to the target application. • Load—Validates and loads data to the target application.

Table 1-16 (Cont.) Financial Management Application Options and Descriptions

Option	Description
Load Method	Select the method for loading a data file into an application. Available load methods: • Replace—Replaces the data in the application with the data in the load file. For each unique combination of Scenario, Year, Period, Entity, and Value in the data file, the Replace option clears all account values from the application, then loads the value from the data file. Note: You may need to create several small files to load a data file using the Replace mode, especially if the data is very large or if the file contains ownership data. An error message is displayed if the file is too large when you try to load it. • Merge—Overwrites the data in the application with the data in the load file. For each unique point of view that exists in the data file and in the application, the value in the data file overwrites the data in the application. Data in the application that is not changed by the data load file remains in the application. If you select the Accumulate In File option in conjunction with the Merge option, the system adds all values for the same point of view in the data file, and overwrites the data in the application with the total. For each unique point of view that is in the data file but does not have a value in the application, the value from the data file is loaded into the application. • Accumulate—Select the Accumulate option to accumulate the data in the application with the data in the load file. For each unique point of view in the data file, the value from the load file is added to the value in the application. Note: Data for system accounts is not accumulated. • Replace by Security—Performs a data load in Replace mode in which only the

Table 1-16 (Cont.) Financial Management Application Options and Descriptions

Option	Description
	members to which you have access are loaded. This option enables you to perform a data load in Replace mode even if you do not have access to all accounts. When you perform the Clear operation for a period in a sub-cube, only the cells to which you have access are cleared. Data, cell text, and line item detail are cleared, but cell attachments are not cleared.
Accumulate in File	You use the Accumulate in File option in conjunction with the Merge and Replace options. When a data load file contains multiple lines of data for the same point of view, this option first accumulates the data in the file, and then loads the totals into the application based on the selected load option. For each unique point of view in the data file, the value from the load file is added to the value in the application. For example, if you have 10, 20 and 30 in the file, 60 is loaded. Select Yes to enable or No to disable.
Has Ownership	If the file that you are loading contains ownership data, you must indicate this option. If you do not select this option and the data file contains ownership or shares data, an error occurs when you load the file. Select Yes to enable or No to disable.
Enable Data Protection	Enables Oracle Hyperion Financial Data Quality Management, Enterprise Edition to protect target-system data from being overwritten during data loads; is based on a specified protection value. Use this option when data is entered into the target system through a method other than FDMEE. Select Yes to enable or No to disable.
Protection Value	The value that is to be protected during the Load step when Enable Data Protection is enabled. The value in this field is to be a dimension value inside of Financial Management across any dimension. For example, enter Protect This .
Protection Operator	Select operator (= or <>). This is used only when the Enable Data Protection option is enabled. The option enables you to state that the data to be protected is equal (=) or not equal (<>) to the "Protection Value."

Table 1-16 (Cont.) Financial Management Application Options and Descriptions

Option	Description
Enable Journal Load	Enables the loading of journal files. You can load working, rejected, submitted, approved, and posted journals as well as standard and recurring journal templates. You cannot load automated consolidation journals because they are created by the consolidation process. The default setting for this option is No. This setting is also used with the Data Value selected for the location on the Location screen to determine when and how data is loaded to Financial Management as journals. Select Yes to enable or No to disable.
Drill Region	Select Yes to create a drill region. Drillable region definitions are used to define the data that is loaded from a general ledger source system and specify the data drillable to FDMEE. In data grids and data forms, after the regions have been loaded, cells that are drillable are indicated by a light blue icon at the top left corner of the cell. The cell context menu displays the defined display name, which then opens the specified URL. A region definition load file consists of the following information: • Scenario, Year, Period, Entity, Account • Display Name (for cell context menu) and URL (to drill to)

Table 1-16 (Cont.) Financial Management Application Options and Descriptions

Option	Description
Enable Cell Text Loading	Enables the loading of text and documents to a data cell. FDMEE archives documents in the EPM_ORACLE_HOME/ products/ FinancialDataQuality/data directory. Select Yes to enable or No to disable.

 Note:

FDMEE does not load multiple cell text to an intersection in Financial Management. If a load using an append mode is run and new cell text is added to an intersection that already has cell text, the old cell text is replaced by the new cell text and not appended.

Enable Consolidation	Enables consolidation in the data load. Consolidation is the process of gathering data from dependent entities and aggregating the data to parent entities. Launching consolidation runs the consolidation process for the specified scenario, year, period, entity, and value. As a part of that process, consolidation for all descendant entities and all prior time periods within the same year is run, if it has not been run previously. The consolidation process runs all calculation rules functions for each affected entity and value, and the translation process runs as necessary to convert from child entities to parent entities. Select Yes to enable or No to disable.

Table 1-16 (Cont.) Financial Management Application Options and Descriptions

Option	Description
Consolidation Type	Select the consolidation type for the data load. Available consolidation types: Impacted—The Consolidate (Impacted Consolidation) option is available for any parent entity with a calculation status of CN or CN ND. When you select this option for a parent entity, the system performs calculations (Calculate, Translate, Consolidate) for any dependent entity within the consolidation path of the selected parent that has a calculation status of CN, CN ND, CH, TR, or TR ND, on the current period or on any prior period in the current year. Consolidate is the most efficient option because only entities that require consolidation are updated. Process units with a status of NODATA on the current period and all prior periods are skipped. Process units with a status of OK or OK SC on the current period are not recalculated, retranslated, or reconsolidated. If the selected parent has a status of CN or CN ND in the prior period, consolidation runs for all periods from the first period in the current year where the parent is impacted until the current period. All with Data—The Consolidate All with Data option is available for any parent entity, regardless of its status. When you select this option for a parent entity, the system consolidates every dependent entity within the consolidation path of the selected parent that contains data, regardless of its status, in the current period or in any of the prior periods. The system also runs calculation rules for the dependent entities. It does not perform a consolidation on entities that contain zeroes or no data. This option is useful for updating system status from OK SC to OK after metadata changes. Process units with a status of NODATA on the current period and all prior periods are skipped. Consolidation Options units with a status of OK or OK SC on the current period are recalculated, retranslated, and reconsolidated. If the selected parent has a status of CN or CN ND in the prior period, consolidation runs for all periods from the first period in the current year where the parent is impacted until the current period.

Table 1-16 (Cont.) Financial Management Application Options and Descriptions

Option	Description
	• All—The Consolidate All option is available for any parent entity, regardless of its status. When you select this option for a parent entity, the system performs calculations for every process unit within the consolidation path of the selected parent, regardless of its status. It consolidates all entities whether they contain data or not. This option is useful when an update from prior periods is required, or when an entity with no data needs to be populated using allocations. This option should be used sparingly because the system does not omit entities with no data, which can have a significant impact on consolidation performance. Process units with a status of NODATA on the current period are calculated, translated, and consolidated. Process units with a status of OK or OK SC on the current period are recalculated, translated, and reconsolidated. If the selected parent has a status of CN or CN ND in the prior period, consolidation runs for all periods from the first period in the current year where the parent is impacted until the current period. • Entity—Calculates the contribution of each entity to its parent, and then totals all contributions to arrive at the consolidated numbers. • Force Entity Only—Forces calculation to run for all selected contribution values to arrive at the consolidation numbers.
Enable Force Calc	Enables the execution of the default calculation call prior to a consolidation run. Select Yes to enable or No to disable.
Enable Force Translate	Enable to force translation to run for all selected cells. Select Yes to enable or No to disable.
Translation Level	Determines the translation level (levels to include for translation) of rows and columns when loading data. Available levels: • <Entity Curr Adjs> • <Entity Curr Total> • <Entity Currency> • <Parent Curr Adjs> • <Parent Curr Total> • <Parent Currency> (Default)
Enable Multi-Load Zero Loading	Select Yes to load 0 values during a multiple period load.

Table 1-16 (Cont.) Financial Management Application Options and Descriptions

Option	Description
Journal Status	To load journals with a specific status, select the status. The journal status indicates the current state of the journal. Available journal statuses: • 1—Working: Journal is created. It has been saved, but it can be incomplete. For example, a label or single entity may need to be assigned. • 2—Submitted: Journal is submitted for approval. • 3—Approved: Journal is approved for posting. • 4—Posted: Journal adjustments are posted to the database (default). • 5—Journal is rejected or unposted.
Journal Default Value	Specify the default value of the journal. The default setting for this option is <Entity Curr Adjs>. Available values are: • [Contribution Adjs] • [Parent Adjs] • <Entity Curr Adjs> • <Parent Curr Adjs>
Journal Enable JV ID per Entity	Assign a journal id (journal numbering) to entities that are being loaded. Select Yes to assign one journal id for each entity in the POV. Select No, to assign one id for all data in the POV. This option is only used when loading journals. The default setting is Yes. This option only applies to FDMEE data that is imported as data and not through the Journal interface. Because regular data that is imported into FDMEE can be loaded to Financial Management as a journal and Financial Management requires all journals to have a JV ID, this option enables FDMEE to determine how the JV IDs¹ are created.
Journal Balancing Attribute	Select the journal balancing attribute used in the journal subsection. Available attributes are: • B—Balanced • U—Unbalanced • E—Balanced By Entity This option is only used when loading journals to Financial Management. The default setting is U-Unbalanced.
Currency Rate Account for Beginning Rate	Specify the Financial Management target application currency rate account for the beginning rate.

Table 1-16 (Cont.) Financial Management Application Options and Descriptions

Option	Description
Currency Rate Account for Ending Rate	Specify the Financial Management target application currency rate account for the ending rate.
Currency Rate Account for Average Rate	Specify the Financial Management target application currency rate account for the average rate.
Source Language for Member Description	Specify the source language for the description when members are loaded. Available languages are shown on the drop-down.
Global User for Application Access	Option to override the Single Sign-On logon to the Financial Management applications. When a user name is specified for this option, this user name is used to access the Financial Management application instead of the FDMEE sign-on user name. Specify the user name of a user who has administrator access to the Financial Management application.

Enabling Intercompany Partner Dimensions for Financial Management

Oracle Hyperion Financial Data Quality Management, Enterprise Edition supports intercompany partner (ICP) dimensions used by Oracle Hyperion Financial Management. The ICP dimension represents a container for all intercompany balances that exist for an account. ICP is a reserved dimension used in combination with the account dimension and custom dimensions to track and eliminate intercompany transaction details.

When working with ICP transaction in FDMEE, note the following:

- ICP transactions can be loaded only from a file.
- If the import format contains a dimension of type "ICPTRANS," then it is considered an ICP Transaction load.
- If the import format does not contain any dimension of type "ICPTRANS," then it is considered a standard data load.
- When ICP transactions are enabled, the following attributes are included:
 - Attr1 – Transaction ID
 - Attr2 – Transaction Sub ID
 - Attr3 – Reference ID
 - Attr4 – Transaction Date
 - Attr5 – Entity Currency Amount
 - Attr6 – Transaction Currency
 - Attr7 – Conversion Rate
 - Attr11 - Comment

ICP transactions are included in the import format.

When registering a Financial Management application, an "Enable ICP Transactions" option is available. When this option is enabled, additional ICP dimensions are added to Dimension Details. These dimensions are registered with dimension classification "ICPTRANS" (ICP Transactions) and the attributes of the values are stored in the data table column name "ATTR." Source values of the ICP dimensions cannot be transformed by mapping. If some transformation is required, then you use a script from another dimension map. Also see [Entity and Intercompany](#).

To enable intercompany partner dimensions:

1. On the **Setup** tab, under **Register**, select **Target Application**.
2. In the **Target Application** summary grid, select a Financial Management application.
3. Select the **Dimension Details** tab.
4. Click **Enable ICP Transactions**.
5. When the Refresh Metadata Confirmation window is displayed, click **Continue and Save Later** to add the ICP transactions.

In this case, to save the ICP transactions, you need to click **Save** to save the transactions with the dimension details.

To add and save the ICP transactions immediately, click **Save**.

6. Click **Save**.

Deleting Registered Target Applications

Use caution when deleting registered target applications. When you delete a registered target application, the target application is removed from the Target Application screen, and all metadata and data rules associated with the application are removed.

To delete a registered target application:

1. On the **Setup** tab, under **Register**, select **Target Application**.
2. In **Target Application**, select the target application, and then click **Delete**.
3. Click **OK**.

When you delete a target application, the application is marked for deletion and is unavailable for any metadata or data rule processes, including the import format and location creation options. All the existing rules involving the target application are removed.

Note:

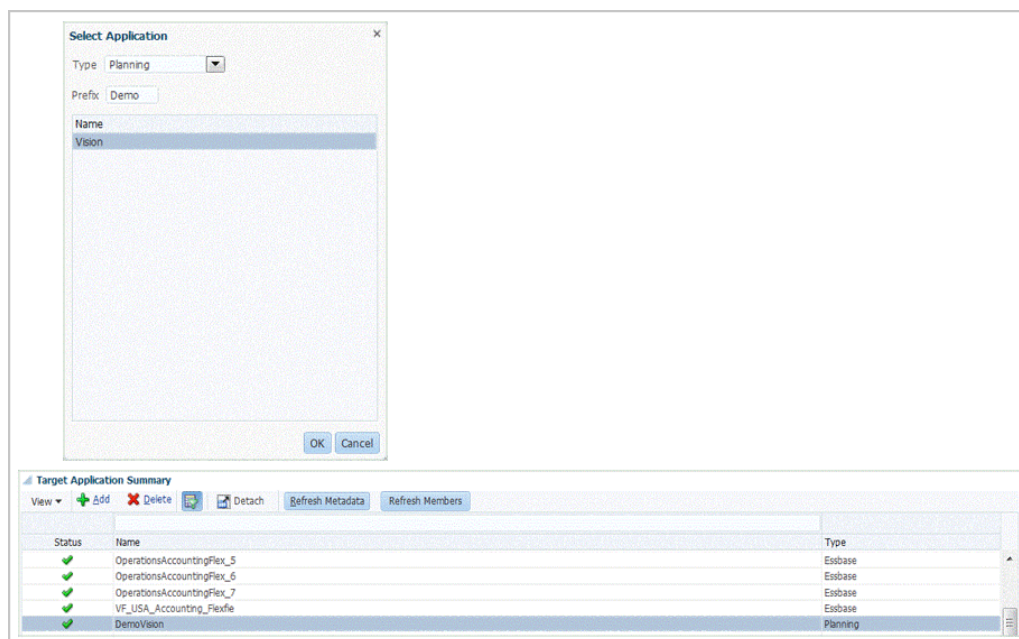
After a target application is deleted and the process has run successfully, use the Target Application screen to set up the same application and redefine the rules.

4. Click **Save**.

Reusing Target Application Names Multiple Times

Oracle Hyperion Financial Data Quality Management, Enterprise Edition supports the registration of target applications with the same name. You might use this feature when you have multiple service environments and the application name is the same in each environment, or the application names are identical in development and production environments. This feature enables you to add a prefix to the application name when registering the application so that it may be registered successfully in FDMEE and can be identified correctly in the list of target applications.

In the following example, the user has selected the target application name "Vision" and prefixed it with the name "Demo." The result is a new target application with the name "DemoVision."



A target application with a prefix is not backward compatible, and cannot be migrated to a 17.10 or earlier release. Only a target application without a prefix name can be migrated to an earlier release.

For information on adding the prefix, see [Step 5 in Registering a Target Application](#).

Selecting Source Accounting Entities

Select accounting entities to specify the Fusion Financials/E-Business Suite ledgers or PeopleSoft business units from which the metadata and/or data is extracted.

After you register and initialize your source systems for use with Oracle Hyperion Financial Data Quality Management, Enterprise Edition, you select the source system accounting entities (ledgers or business units) to use for integration. Segments and chartfields are the terminologies specific to Fusion, E-Business Suite, and PeopleSoft.

To select source accounting entities in a source system:

1. On the **Setup** tab, under **Register**, select **Source Accounting Entity**.
2. In **Source Accounting Entities**, select the **Entities** tab.

3. Select the source system type.
4. Select a source system.

The source accounting entities are displayed in a grid. You can click the column header to sort any column. The Entities tab displays the following columns:

For Fusion and E-Business Suite source systems:

- **Select**—A check mark indicates that the accounting entity (ledger) is available.
- **Accounting Entity**—Name of the ledger
- **Chart of Accounts**—Chart of accounts name
- **Currency**—The functional currency of the Fusion and E-Business Suite ledger
- **Calendar**—The Fusion and E-Business Suite ledger calendar. The Fusion and E-Business Suite ledger is a collection of chart of accounts, currency, and calendar. For example, 4–4–5, Weekly, Accounting, and so on.
- **Responsibility Name**—Displays the general ledger drill-through responsibility. The drill through responsibility must be set in FDMEE to enable users to drill through to E-Business Suite. The responsibility selected must have the authority to view summary journals and journal details for the selected ledger in the E-Business Suite.

For PeopleSoft source systems:

- **Select**—Select the check box to make the business unit available.
 - **Business Unit**—Business unit name
 - **Currency**—The base currency for the business unit
5. For Fusion and E-Business Suite source systems, select the general ledger **Responsibility Name**. See [Assigning General Ledger Responsibility](#).
 6. For each ledger or business unit that you want to make available in FDMEE, select the check box.
 7. Click **Save**.

**Tip:**

To filter by the business unit or accounting entity, ensure that the filter row is displayed above the column headers. (Click the  to toggle the filter row.) Then, enter the text to filter.

Assigning General Ledger Responsibility

In the E-Business Suite General Ledger, the system administrator assigns users general ledger responsibility. General ledger responsibility provides the authentication required for Oracle Hyperion Financial Data Quality Management, Enterprise Edition to drill through to the Fusion and E-Business Suite journal summary page.

To assign general ledger responsibility:

1. In **Source Accounting Entities**, in the **Responsibility Name** column, click  to select a **General Ledger Responsibility**.

The Responsibility Name is the Responsibility ID the user is logged in under when drilling through to the source system. Select the responsibility name *only* if you want to drill through; otherwise, leave it blank.

2. Repeat the above process as necessary for all selected ledgers in the source system.
3. Click **Save**.

Working with Source Accounting Entity Groups

An accounting entity group is a logical grouping of common accounting entities (for example, the same Chart of Accounts, the same Calendar, or the same currency). Accounting entity groups are used to extract data from multiple accounting entities in a single data rule execution. They facilitate sharing the same data because multiple accounting entities can belong to more than one accounting entity groups.

To work with accounting entity groups, note the following:

- Accounting entity groups can be used only with data rules.
- When a data rule in a location includes an accounting entity, then the rule is constrained by the accounting entity in the definition. In this case, the data rule in the location cannot use an accounting entity group.
- When a data rule in a location has no accounting entity, then an accounting entity or an accounting entity group must be specified in the definition. The data rule execution extracts data from a single accounting entity or from all the accounting entities in an accounting entity group.
- Data load to write-back rules accept only an accounting entity (and not an accounting entity group) in their definition.
- Metadata rules accept only an accounting entity (and not an accounting entity group) in their definition.
- Oracle Hyperion Financial Data Quality Management, Enterprise Edition does not enforce that entities belong to the same Chart of Accounts.

You view and maintain accounting entity groups using the Entity Groups tab in the Source Accounting Entities feature. The Entity Groups tab consists of two regions: Entity Groups, to which you can add a new group, and the Entity Groups Entities, from which you can add accounting entities.

To add a source accounting entity group:

1. On the **Setup** tab, under **Register**, select **Source Accounting Entity**.
2. In **Source Accounting Entities**, select the **Entity Groups** tab.
3. Select the source system type.
4. In the **Entity Groups** grid, click **Add**.
Blank Name and description rows are added at the top of the grid.
5. In **Name**, enter the name of the accounting entity group.
6. In **Description**, enter a description of the accounting entity group.
7. In the **Entity Group Entities** grid, select the accounting entities to add.

The source accounting entities are displayed in the Entity Group Entities grid. You can click the column header to sort any column. The Entity Group Entities grid displays the following columns:

For Fusion and E-Business Suite source systems:

- **Select**—A check mark indicates that the accounting entity (ledger) is available.
- **Accounting Entity**—Name of the ledger
- **Chart of Accounts**—Chart of accounts name
- **Currency**—The functional currency of the Fusion and E-Business Suite ledger
- **Calendar**—The Fusion and E-Business Suite ledger calendar. The Fusion and E-Business Suite ledger is a collection of chart of accounts, currency, and calendar. For example, 4–4–5, Weekly, Accounting, and so on.
- **Responsibility Name**—Displays the general ledger drill-through responsibility. The drill-through responsibility must be set in FDMEE to enable users to drill through to E-Business Suite. The responsibility selected must have the authority to view summary journals and journal details for the selected ledger in the E-Business Suite.

For PeopleSoft source systems:

- **Select**—Select the check box to make the business unit available for the accounting entity group.
- **Business Unit**—Business unit name
- **Currency**—The base currency for the business unit

8. Click **Save**.

Loading Excel Data

Use the Load from Excel feature to load data to and from a Microsoft Excel workbook. This feature enables you to load large amounts of application information and entity types without having to enter each line manually, and to push data into tables without SQL access.

Available options include:

- **Download to Excel**—Select an entity and download the data from the corresponding table to an Excel spreadsheet.
- **Upload from Excel**—Import source data representing one more ranges from an Excel spreadsheet.

Caution:

When loading data using this method, Oracle Hyperion Financial Data Quality Management, Enterprise Edition does not validate the data. It is the responsibility of the user to validate the date using this method.

Downloading to Excel

You can select an Oracle Hyperion Financial Data Quality Management, Enterprise Edition entity and download the data from the corresponding table to an Excel spreadsheet.

When downloading to Excel, note that the format of the Excel file must include:

- table name
- Column names in the second row
- Data from the table

The following example shows how a Period entity is mapped in Excel:

TPOVPERIOD					<- Table Name
PERIODKEY	PRIORPERIODKEY	PERIODDESC	PERIODTARGETM	YEARTARGET	<- Column Name
1/1/2013	12/1/2012	Jan-13	Jan	2013	<- Data
2/1/2013	1/1/2013	Feb-13	Feb	2013	
3/1/2013	2/1/2013	Mar-13	Mar	2013	

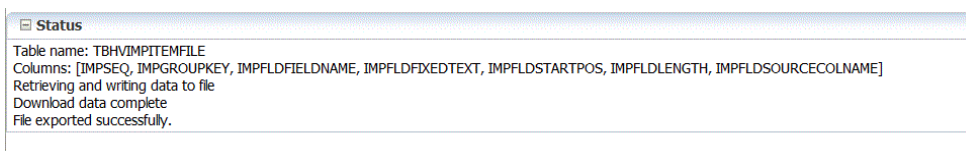
To download data to an Excel spreadsheet:

1. On the **Setup** tab, under **Integration Setup**, select **Excel Interface**.
2. From **Excel Interface**, then **Download to Excel**, and then from **Entity Type**, select the **FDME** entity from which to download data.

Available entity types:

- Application Category Mapping
- Application Period Mapping
- Batch Definition
- Batch Groups
- Batch Jobs
- Category Mapping
- Check Entity Group
- Check Entity Item
- Check Entity Detail
- Check Entity Header
- Data Rule
- Data Rule Parameters
- Import Format Detail-Adapter
- Import Format Detail-ERP
- Import Format Detail-File
- Import Format Header
- Location

- Logic Group Detail
 - Logic Group Header
 - Period Mapping
 - Source Period Mapping
 - User Setting
 - Other (prompts for table name)
3. **Optional:** If you selected an **Other** entity type, enter the name of the table in the **Table Name** field.
 4. In **File**, enter the name of the Excel file to which to download the data.
You can also download a file by clicking **Select**, navigating to the file on the Select screen, entering the new file name, and then clicking **Download**.
 5. Click **Download**.
When the Excel spreadsheet is uploaded, the names of the tables and columns that have been downloaded are shown, and the message: "File imported successfully" is displayed.



6. **Optional:** To open the downloaded Excel file, next to the File name field, click **Select**.
From the **Select** screen, choose the Excel spreadsheet and click **Download**.
Save or open the Excel spreadsheet.

Uploading from Excel

An Excel spreadsheet that is used as an import source represents one or more ranges. The first cell of a range contains the name of the Oracle Hyperion Financial Data Quality Management, Enterprise Edition table to be imported. The second row of a range contains the column names of the table identified in the first row. Rows 3 and higher contain the data to be loaded to FDMEE.

Range names begin with the FDMEE import identifier *ups*, for example, *upsCategory*. When imported tables are related, the parent table must be imported prior to the child table. Range names process in alphabetical order. To ensure the correct parent-child order, assign range names such as *upsAParent* and *upsBChild*.

Also note the following behavior when uploading data from Excel:

- Data is only inserted. It cannot be updated or deleted.
- Data is not validated.
- When FDMEE encounters a duplicate row, the row is skipped.

To upload data from an Excel spreadsheet:

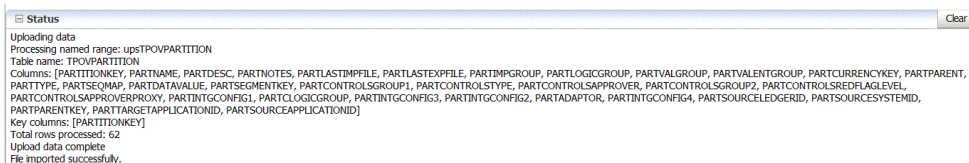
1. On the **Setup** tab, under **Integration Setup**, select **Excel Interface**.

2. From **Excel Interface**, and then **Upload from Excel**, in **File**, enter the name of the Excel file to upload.

You can also select a file by clicking **Select**, navigating to the file on the Select screen, and clicking **OK**.

3. Click **Upload**.
4. **Optional:** To open the downloaded Excel file, next to the **File** name field, click **Select**.
5. **Optional:** You can also click **Upload** and browse to and select an Excel spreadsheet from an alternate directory.

When the Excel spreadsheet is uploaded correctly, the Status pane shows the processed ranges, and the message: "File imported successfully" is displayed.



2

Integrating Tasks

Related Topics

- [Working with Import Formats](#)
- [Defining Locations](#)
- [Defining Period Mappings](#)
- [Defining Category Mappings](#)

Working with Import Formats

Import formats define:

which fields (columns) are extracted from the source system and how the data is stored in the Oracle Hyperion Financial Data Quality Management, Enterprise Edition staging table.

the data format and mapping information for columns in the Oracle Hyperion Planning source system and target applications for a source adapter based integrations.

- mapping between the source system segments or chartfields and the dimensions (used in the metadata load and data load process).
- mapping between the EPM dimensions and the source system segments or chartfields (used in the write-back process).
- mapping information for source adapter based integrations
- mapping information between source and target EPM dimensions for data synchronization.

Note:

You do not need to define import mappings for use with human resources source systems. This is only applicable to the rebuilt Peoplesoft HCM integration.

Import formats are created for a single accounting entity. However, if you are importing data from multiple accounting entities that have the same Chart of Accounts, you can define one Import Format using a representative accounting entity, and then use it for importing data for all accounting entities with the same Chart of Accounts.

You define import formats to map source segments or chartfields to target dimensions. You can create import formats to:

You work with import formats on the Import Format screen, which consists of three sections:

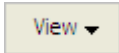
- Import Format Summary—Displays common information relevant to the source and target applications.
- Import Format Detail—Enables you to add and maintain import format information.
- Import Format Mappings—Enables you to add and maintain import format mapping information.

Defining the Import Format

Use the Import Format summary section to view, add, and delete import format summary information.

Viewing Import Format Information

Table 2-1 Import Format View Options and Descriptions

View Option	Description
	Customizes views. Options include: • Columns—Select the columns to display in the data grid including: – Show All – Name – Source System – Accounting Entity – Source Adapter – Target Application • Reorder Columns—Use to change the order of the columns. When you select this option, the Reorder Columns screen is displayed. You can select a column, and then use the scroll buttons on the right to change the column order. • Detach/Attach—Detaches columns from the data grid. Detached columns are displayed in their own window. To return to the default view, select View, and then click Attach or click Close. • Query by Example—Use to toggle the filter row. You can use the filter row to enter text to filter the rows that are displayed for a specific column. To clear a filter, remove the text to filter by in the text box, and then the press [Enter] key. All text is case sensitive.

Adding Import Formats

You work with import formats on the Import Format screen, which consists of three sections:

- Import Format Summary—Displays common information relevant to the source and target applications.
- Import Format Detail—Enables you to add and maintain import format information.

- **Import Format Mappings**—Enables you to add and maintain import format mapping information.

To add an import format for an Enterprise Resource Planning (ERP) source:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. In the **Import Format** summary task bar, select **Add**.
In the upper grid of the Import Formats screen, a row is added.
3. In **Name**, enter a user-defined identifier for the import format.
You cannot modify value in this field after a mapping is created for this import format.
4. In **Source**, select the source system.
5. In **Target**, select the target system.
6. **Optional:** In **Concatenation Character**, specify the character to use for concatenating two segments when multiple segments are specified for sources.

Set the concatenation member by specifying the concatenation character.

For non-ARM data loads and metadata loads, the underscore character “_” is used to concatenate two segments. For ARM data loads, you cannot use an underscore character with a mapping that uses the “Like” method. Therefore, for ARM mappings, specify a different concatenation character for the data load mapping details.

7. In **Description**, enter a description of the import format.
8. In **Accounting Entity**, select the accounting entity.
For Fusion and E-Business Suite source systems, the accounting entity is the ledger. For PeopleSoft source systems, the accounting entity is the business unit.
9. Define the import mappings from source to target.
See [Defining the Import Format Mappings](#).

For information on one-to-one mappings and many-to-one mappings, see [Defining Import Format Mappings](#).

To add an import format for a source adapter (open interface adapter):

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. On the **Import Format** summary task bar, select **Add**.
In the upper grid of the Import Formats screen, a row is added.
3. In **Name**, enter a user defined identifier for the import format.
You cannot modify the value in this field after mapping is created for this import format.



Note:

Do use non-ASCII characters in an import format name when the import source is an adapter.

4. In the **Description**, enter a description of the import format.

5. In **Source**, select the source for the source adapter.
6. In **Source Adapter**, select the source adapter name.
7. In **Target**, select **EPM** and select the target system.
8. In **Drill URL**, enter the URL that identifies the URL to use for drilling through when using this import format.
9. Define the import mappings from source to target. See [Defining the Import Format Mappings](#).

For information on one-to-one mappings and many-to-one mappings, see [Defining Import Format Mappings](#).

To add an import format for a file-based data load system:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. In the **Import Format** summary task bar, select **Add**.
In the upper grid of the Import Formats screen, a row is added.
3. In **Name**, enter a user-defined identifier for the import format.
You cannot modify the value in this field after a mapping has been created for this import format.
4. In **Description**, enter a description of the import format.
5. In **Source**, select **File** for the source system.
6. From **File Type**, select the format of the file.
 - Fixed - Numeric Data
See [Defining Import Formats for File-Based Mappings](#).
 - Delimited - Numeric Data
See [Defining Import Formats for File-Based Mappings](#).
 - Multi Column - Numeric Data
See [Loading Multi-Column Numeric Data](#).
 - Delimited Data - All Data Type
See [Setting the Import Format Data Types](#).
 - Fixed Data - All Data Type
See [Setting the Import Format Data Types](#).
 - Multi Column - All Data Type
7. If the file type is "delimited," in the **File Delimiter** field, select a type of delimiter.
Available delimiter symbols:
 - comma (,)
 - exclamation (!)
 - semicolon (;)
 - colon (:)
 - pipe (|)
 - tab

- tilde (~)
8. In **Target**, select the target system.
 9. In **Drill URL**, enter the URL used for the drill-through.

To display the **Drill URL** editor, click .

10. In the **Mapping** section, map any dimensions.
11. Click **Save**.

To define import formats for file-based mappings, see [Defining Import Formats for File-Based Mappings](#).

Import Formats and Data Load Rules

Depending on the source and target types that you select in import formats, different types of data load rules can be created.

Table 2-2 Data Load Rule

Source System	Target System
ERP (Enterprise Resource Planning)	EPM (Enterprise Performance Management)

Table 2-3 Write-Back Rule

Target System	Source System
EPM	ERP

Table 2-4 Data Synchronization

Source System	Target System
EPM	EPM

Deleting an Import Format

To delete an import format:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. In **Import Format**, from the **Import Format** summary grid, select the import format, and then click **Delete**.
3. In **Delete Import Format**, click **OK**.

Querying by Example

You can filter the import formats in the Import Format summary section using the Query by Example feature. To filter by Import Format Name, ensure that the filter row is displayed above the column headers.

To query by example:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.

2. In **Import Format**, on the **Import Format** task bar, select . A blank row is displayed above the column headers.
3. Enter text to filter the rows that are displayed.

Defining Import Format Mappings

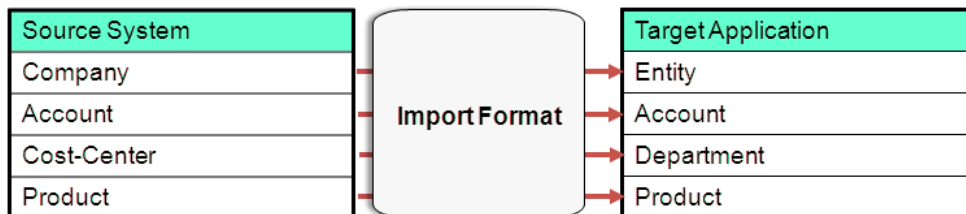
The import format identifies the content structure of the source file or system. When you create an import format, you define the settings and the import format mapping definition between the source and target system dimensions. You can create:

- **One-to-One Mappings**—Single Segment and Single Chartfield mappings:
 - Define a simple one-to-one mapping between source segments and target dimension members.
 - Pull all members from the source value set as members into the target dimension.

Note:

The topic is unavailable for file-based data load definitions.

The following figure shows a one-to-one mapping between segments or chartfields in a source system and dimensions in a target EPM application.



Note:

Each member in the source is created as a single member in the target dimension.

Note:

When importing a mapping file that has maps for one dimension, the dimension name needs to be in the first field of the file.

- **Many-to-One Mappings**—You can concatenate segments or chartfields to map multiple segments or chartfields from the source into a single EPM dimension member.

When dimensions are mapped as a concatenated segment, the new dimension is created based on the traversal order that you define for the source hierarchies into the concatenated member target dimension. Unlimited number segments may be concatenated into the target dimension. The following table shows how the segments map to dimensions. In this example, two segments (Company and Department) map to one dimension (Entity).

Table 2-5 Segment Mappings to EPM Dimensions

Segment/Chartfield	EPM Dimension
Company	Entity
Department	Entity
Product	Product
Account	Account

- Concatenations of any number of source segments for Enterprise Resource Planning (ERP) sources like Oracle E-Business Suite and PeopleSoft.

Defining the Import Format Mappings

When you define an import format, you define the import format mappings for the metadata and data rule from the Enterprise Resource Planning (ERP) source system.

To define the Enterprise Resource Planning (ERP) system import format mappings:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. From the **Import Format Summary** section, select an import format.
3. Go to the **Import Format Mapping** section.
4. In **Add**, select the target.
5. From **Source Column**, specify the source column from which to map.

For Fusion and E-Business Suite source systems, you can concatenate segments to map multiple segments from the source into a single EPM target dimension. The drop-down list displays all of the source system segments for the accounting entity that you selected when you defined the import format details.

Note:

Verify your entries and ensure that, if you enter Segment 1, Segment 2, and Segment 3 that, the segments are displayed in sequence. For example, Segment 3 should not come before Segment 1. Entering segments out of sequence can cause an error.

For Oracle Hyperion Financial Management applications, concatenate segments are *not* available if your target dimension is an Entity dimension type. Typically, the intercompany segment in Fusion and E-Business Suite or affiliate in PeopleSoft is related to a company or business unit. If concatenation was enabled, you could not determine the ICP dimension value.

For PeopleSoft source systems, select chartfields as necessary. The drop-down list displays the source system chartfields for the Accounting Entity (business unit) that you selected for the import format.

When the target application is an ARM data load type, you can set the concatenation member by specifying the concatenation character.

For non- ARM data and metadata loads, the underscore character “_” is used to concatenate two segments. For ARM data loads, you cannot use an underscore character with a mapping that uses the “Like” method. Therefore, for ARM mappings, specify a different concatenation character for the data load mapping details.

6. For EPM and file-based source systems, you can add expressions in the **Expression** field. See [Adding Import Expressions](#).
7. Click **Save**.

Concatenating Source Dimensions for Enterprise Resource Planning (ERP) Segments

The Import Format enables you to concatenate Enterprise Resource Planning (ERP) segments and define mappings on the concatenated code combinations. Concatenation facilitates a simple set of data load mappings with an unlimited number of segments that can be concatenated.

The order of concatenation is based on the order columns are inserted.

For Oracle E-Business Suite metadata loads you load the concatenated dimension and the load does not require hierarchies to be associated with it. PeopleSoft sourced applications handle metadata loads differently than E-Business Suite. For PeopleSoft source applications you must have hierarchies associated with the concatenated dimension. This is a requirement for PeopleSoft metadata loads.

To concatenate a source dimension:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. In the **Import Format** summary task bar, select the import format.
3. In the **Import Format Detail** section, from the **Concatenation Character**, select the concatenation character.

For non-ARM data loads and metadata loads, the underscore character “_” is used to concatenate two segments. For ARM data loads, you cannot use an underscore character with a mapping that uses the “Like” method.
4. In **Data Load Mapping**, click **Add**.
5. Select **Dimension Row**, and then the *target dimension*.
6. In **Source Column**, select the first source column to concatenate to the target.

For example, to have two source columns (Company and Account separated by a dash), select the Company source columns.
7. In **Data Load Mapping**, click **Add**.
8. In **Source Column**, select the second source column to concatenate to the target.

**Note:**

Only EPM source types enable you to specify a value in the Expression field.

9. Repeat steps 4-8 for each source column to concatenate.

KSESSB_EBS: Mappings

Data Load Mapping Write Back Mapping

Add ✖ Delete

Source	Target Dimension
Ledger	Organization
Fund	Organization
Organization	Organization

10. Click **Save**.

Adding Import Expressions

Oracle Hyperion Financial Data Quality Management, Enterprise Edition provides a set of powerful import expressions that enable it to read and parse virtually any trial balance file into the FDMEE database. Advanced expressions are entered into the Expression column of the field grid of the Import Formats screen. Import expressions operate on the value that is read from the import file.

Also see [Stacking Import Expressions](#) and [Processing Order](#).

To add an import format expression:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. In **Import Format**, from the **Import Format Mapping** grid, select the file-based source column.
3. In **Expression**, specify the import expression.
4. **Optional:** You can also specify the expression type and value on the **Add Expression** field.
 - a. Click .
 - b. In **Add Expression**, under **Expression Type**, select the expression type.
The number and types of expressions available depend on the field that is being modified (for example, Account or Account Description).
 - c. In **Expression Value**, enter the value to accompany the expression and click **OK**.
5. In **Import Format Mapping**, click **OK**.

Import Expression Types

Expressions supported by Oracle Hyperion Financial Data Quality Management, Enterprise Edition:

- [Nonstandard Numeric Sign Conventions](#)
- [Converting from European to U.S. Notation](#)
- [Padding Fields with Leading Fills](#)
- [Padding Fields with Trailing Fills](#)
- [Multiplying by Whole-Number and Decimal Factors](#)
- [Disabling Zero Suppression](#)

Nonstandard Numeric Sign Conventions

The Sign expression is used to manage nonstandard numeric sign conventions. Oracle Hyperion Financial Data Quality Management, Enterprise Edition interprets numbers with leading and trailing minus signs and numbers within parentheses as negative numbers. You can also use other leading and trailing characters to indicate negative numbers. To define custom signs, use expressions that follow this form: Sign=[Positive String],[Negative String].

For example, if positive numbers are followed by DR (1,000.00DR), and negative numbers are followed by CR (1,000.00CR), the expression is Sign=DR,CR.

Numbers within <> are also treated as negative. For example, if you specify (100.00) and <100.00> both are treated as negative numbers.

If positive numbers are unsigned (1,000.00), and negative numbers are followed by CR (1,000.00CR), the expression is Sign=,CR.

Displaying Debit and Credit Columns

The DRCRSplit expression is used to parse split numeric columns. By default, Oracle Hyperion Financial Data Quality Management, Enterprise Edition assumes that numeric values in Amount fields are debits. However, you can position debit values on the left and credit values on the right.

Table 2-6 DRCRSplit expression example

Account Number	Description	Debit	Credit
1000-000-00	Cash-Operating Account	68,603.91	
1010-000-00	Cash-FANB-AP		177,216.16

DRCRSplit, which enables FDMEE to interpret left-right positioning and to assign the correct sign, follows the format DRCRSplit=Mid Point of the DR and CR columns.

When the file is imported, credit amounts are assigned negative signs (and thus are interpreted as positive), and debit amounts are unchanged (and thus are interpreted as negative).

Converting from European to U.S. Notation

The `Fill=EuroToUS` expression is used with the Amount field to trigger a number format conversion from (.,) to (.) format.

Padding Fields with Leading Fills

The `Fill=LeadingFill` expression is used to fill fields with leading characters. Text values that are shorter than the specified fill expression are padded, as directed by the fill expression.

Padding Fields with Trailing Fills

The `Fill=TrailingFill` expression is used to fill fields with trailing characters. Text values that are shorter than the specified fill expression are padded, as directed by the fill expression. For example, if the account number is 103950- and the expression `Fill=0000000000` is used, the account number after import is 103950-000. Another example is an account number of 243150 with the expression `Fill=111111111`. The account number after import is 243150111.

Multiplying by Whole-Number and Decimal Factors

The `Factor=Value` expression is used to factor source-file amounts by user-defined values.

Amounts can be multiplied by whole numbers or decimals; thus, for example, amounts can be doubled or halved.

Disabling Zero Suppression

The `NZP` expression is used to disable zero suppression during the data-load process. By default, Oracle Hyperion Financial Data Quality Management, Enterprise Edition bypasses accounts in the trial balance that have zero balances. In certain circumstances, you may want to load all accounts, to ensure that values that should be zero are replaced. You enter `NZP` in the Expression column of the Amount field to disable zero suppression.

Stacking Import Expressions

One field can use multiple import expressions. To stack expressions, separate the expressions with a semicolon. Consider the order in which the layered expressions are processed.

For example, to stack the import expression for the General Ledger Amount column, enter the expression: `Script=ParseAcct.py;Fill=00000000`.

Defining a Multiple Period Data Load in Import Formats

When you need to load data for an entire fiscal year or multiple years (such as for budgeting and forecasting data); you can use a multi-period file to load the data. With a multi- period file-based load, you can import one single file that has data for multiple periods in multiple columns.

To define a multiple period data load:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. In **Import Format**, from the **Import Format** summary grid, select the file in which to define a multiple period data load.
3. In **File Type**, select **Multi-Period - Numeric Data**.

MGPBCS_Multiperiod: Details

Name: MGPBCS_Multiperiod

Description:

Source Type: ERP

Source: File

* File Type: Multi Period - Numeric Data

Drill URL:

Target Type: EPM

Target: MGPBCS

* File Delimiter: Comma

4. In **File Delimiter**, select the delimited based on the source file.
5. Add column mappings by mapping the source files column numbers with the target application dimensions.

MGPBCS_Multiperiod: Mappings

Source Column	Field Number	Expression	Add Expression	Target
Account	1			Account
Amount	4			Amount
Custom1	2			Custom1
Entity	3			Entity
				Version

6. From the Expression column for **Amount**, click the **Add Expression** editor.

Add Import Format Mapping Expression

* Expression Type: Column=start,end

* Expression Value: 6,17

Description: Use this expression to specify number of amount columns

Example -> 11,22 -> Columns 11 through 22 will be considered amount columns

7. From **Add Import Format Mapping Expression** screen, and then from the **Expression Type** drop-down, select **Column=start,end**.
8. From **Expression Value**, enter the column number of the first amount in the source file, and then the column number of the last amount in the source file. For example, if the source file has forecasting values from January to December, and the amount for January begins from column 6, the December amount is at 17. In this case, when you specify 6, 17, the Expression Value includes the amounts beginning from column 6, and then load value for the next 12 months up to December.
9. Click **Save** to save the import format.
10. **Optional:** If necessary, create a location that includes the import format for the multi-periods.
For more information, see [Defining Locations](#).
11. **Optional** If necessary, create a data load rule.

For more information, see [Defining Data Load Rules to Extract Data](#).

12. From **Data Load**, then **Data Load Rule**, click **Execute** to execute the data load rule.

For more information, see [Running Data Load Rules](#).



Note:

Locations with multi-period import formats cannot be executed from the Data Load Workbench. They are executed using the Data Load Rule only.

Processing Order

For all fields except the Amount field, Oracle Hyperion Financial Data Quality Management, Enterprise Edition processes stacked expressions in the following order:

1. Script
2. Fill or FillL

For the Amount field, FDMEE processes stacked expressions in the following order:

1. DRCRSplit
2. Fill=EuroToUS
3. Script
4. Sign
5. Scale
6. NZP

Defining Import Formats for File-Based Mappings

When the source is a file-based data load, you define the settings and the import format mapping definition, which determines which fields (columns) are extracted from the source system and how data is stored in the staging table.

For detailed information about working with import formats, see [Working with Import Formats](#). For information about working with all data types data loads in import formats, see [All Data Types Data Loads](#).

You can add a mapping row to the import format definition based on the following options:

- Skip
- Currency
- Attribute
- Description
- Dimension Row

Table 2-7 Import format definition options

Import Definition Option	Description
Skip	The skip option is used to indicate rows in the input file that should be skipped. For example, rows with no data, negative numbers, or for specific accounts. The specification for a skip row is defined in the same way as that for a data row, and the system looks for the exact textual match for text entered in the expression field in the indicated location in the input file. The system automatically skips rows in the input file that have "spaces" and "non-numeric" characters in the amount location, so a skip specification is only needed when non-amount data is present in a row of the input file in the same location as the amount. For example, the input file may contain a label named "date." Add an entry for a skip row to indicate the starting column of the text "date," the length of the text, and the exact text to match. The Skip row option is available for both fixed and delimited file types.
Attribute	The TDATESEG table includes 13 attribute columns that you can load from the selected input file. You can provide the location of the attribute in the input file by specifying the starting location and length, or plug the value during processing by entering the value for the attribute in the expression field. If the value for the attribute is entered in the expression field, then a starting location and length are not needed. The attribute fields are generally used to help compose a drill-through URL or for history or documentation needs. You may want to populate the attribute field to support searching and filtering in the Data Load Workbench.
Description	The TDATESEG table includes two description columns, and you can load these columns in the same way as the attribute columns. You can specify the location in the input row that contains a description, or specify an explicit value by entering it in the expression field in the mapping table.

Table 2-7 (Cont.) Import format definition options

Import Definition Option	Description
Currency	Oracle Hyperion Financial Data Quality Management, Enterprise Edition supports the ability to load data that is of a currency different from the default currency of the selected location. This option enables you to specify the location in the input line that specifies the currency for the related amount field. For the file import format, specify a currency on each row of data, or make sure that a currency is specified in the location that uses the selected import format.
Dimension	FDMEER supports multiple entries for a dimension in the import format when the dimension specification is spread between multiple locations on the same line. This feature enables you to concatenate fields for file-based data. To use this option, select the dimension, start and end positions, and the expression.

 Note:

You may encounter issues with loading data if the currency is not specified correctly.

To define an import format for numeric data files with a fixed length:

 Note:

For information about defining import formats for fixed length all data type data files, see [Setting the Import Format Data Types](#).

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. In **Import Format**, from the **Import Format** summary grid, select the file.
3. In the **Import Format Detail** grid, select the type or row to add from the **Add** drop-down.

Available options:

- Skip Row
- Currency Row
- Attribute Row

- Description Row
 - Dimension Row
4. In the **Start** field, specify where on the file the column starts.
 5. In the **Length** field, enter the length of column.
 6. In the **Expression** field, enter the expression that overwrites the contents of the column.

When entering a constant, enter a starting position and length. Use a start position of "1 " and a length of "1."

See [Adding Import Expressions](#).
 7. Click **Save**.

To define an import format for delimited numeric data files:



Note:

For information about defining import formats for delimited all data type data files, see [Setting the Import Format Data Types](#).

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. In **Import Format**, from the **Import Format** summary grid, select a file.
3. In the **Data Load Mappings** section, click **Add**, and select the type or row to add.

Available options are:
 - Currency Row
 - Attribute Row
 - Description Row
 - Dimension Row
4. In **Field Number**, enter the field to import.

The Expression field ignores the Field Number value when you provide a column expression for a multiple period data load.
5. In the **Expression** field, enter the expression that overwrites the contents of the field.

See [Adding Import Expressions](#).
6. Click **Save**.

Concatenating Source Dimensions for a File-Based Source

The Import Format enables for the concatenation of file-based dimensions as the source of target application dimensions. Concatenation facilitates a simple set of data load mappings with an unlimited number of fields that can be concatenated.

To concatenate a file-based source dimension:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.

2. In the **Import Format** summary task bar, select the import format of the file.
Source columns for both fixed and file delimited files can be concatenated.
3. In **Data Load Mapping**, click **Add**.
4. Select **Dimension Row**, and then specify the *target dimension*.
5. In **Source Column**, select the first source column to concatenate.
6. In **Field Number**, enter the field number from the file to import (defaults to the field number from the file when text is selected).
7. In **Data Load Mapping**, click **Add**.
8. Select **Dimension Row**, and then specify the same *target dimension* as in step 4.
9. In **Source Column**, select the second source column to concatenate.
10. Repeat steps 5-6 for each source column to add to the concatenation.
11. To use a concatenation character between source columns:
 - a. Click **Add**, and then specify the *target dimension*.
 - b. In **Source Column** field, enter a name to identify the concatenation character.
For example, enter **Delimited**.
 - c. In **Field Number**, enter **1** or greater.
Do not enter **0** when specifying the field number for the concatenation character column
 - d. In **Expression** field, enter the concatenation character used to separate concatenated source columns.
For example, enter a , (comma) or a . (period).
12. Click **Save**.

Source Column	Field Number	Expression	Add Expression	Target
Account	1			Account
Delimiter	1	.		Account
SubAccount	2			Account

Using the Import Format Builder

When the source system is a file-based data load, use the Import Format Builder feature to map source fields graphically to target dimension members instead of typing the starting position and the field length. This feature is available for both fixed and delimited files.

Note:

The Import Format Builder does not support tab delimited files.

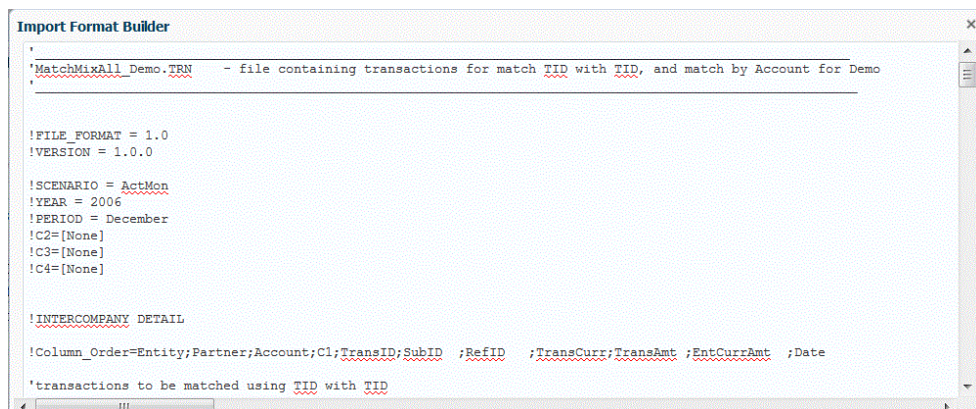
To assign an import format using the Import Format Builder:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. In the **Import Format** summary grid, select a file-based source system.
3. In the **Import Format Mapping** detail grid, click **Build Format**.
4. In **Select file to Load**, locate and choose the file to import, and then click **OK**.

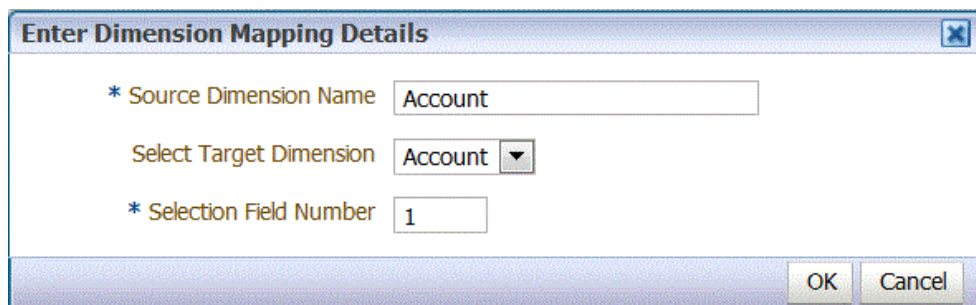
In addition to selecting a file to load, you can:

- Create a new folder under the *inbox* directory.
- Delete a file.
- Download a file
- Upload a file

The file contents are shown on the Import Format Builder screen.

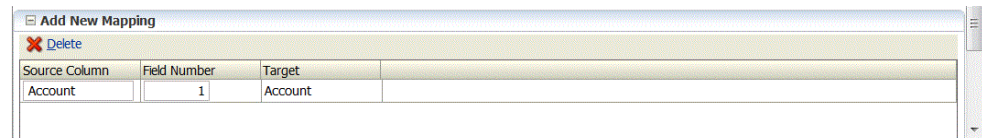


5. Select the text to import.
6. In **Assign selected text as Source Dimension Name**, select **N** to specify the source dimension name on the Enter Dimension Mapping Detail screen.
Select **Y** to use the selected text as the source dimension name and to use the one-to-one mapping between the source dimension and target dimension members.
7. To change dimension mappings details, click **Assign Dimension**.
You must add or change dimension details if you selected **N** in **Assign selected text as Source Dimension Name**.



8. On **Enter Dimension Mapping Details**, from **Source Dimension Name**, specify the name of source dimension to which to assign the selected text.
9. In **Select Target Dimension**, select the target dimension to which to map the source dimension.
10. In **Selection Field Number**, specify the field number from the file to import (defaults to the field number from the file when text is selected).
11. Click **OK**.

When you assign or change source dimension information, the information is shown in the Add New Mapping summary grid.



Source Column	Field Number	Target
Account	1	Account

All Data Types Data Loads

Using the all data type load method, you can load data files that support the following data types to Oracle Hyperion Planning:

- numbers
- text
- Smartlists
- Date

Additionally, you can load data files that include multiple balance columns for a selected calendar period.

Note:

The All Data Type with Security loads only to the currency specified in the import.

All Data Types Data Load Process Description

At a high level, when working with the all data types load method, note the following:

1. In the Target Application feature, on the Application Options tab, select the (either "All data types with auto-increment of line item" or "all data types with security" load method.
2. In the Import Format option, select the file type for the All Data Type load method:
3. For multi-column types only in the Data Load Rule, assign any driver dimension member(s) for multi column data types. The driver dimension is the member into which data is loaded. The driver dimension is only applicable to the multi-column type, not delimited or fixed. If no header record or member expressions are assigned, you can select the members from the Member Selector screen in the import format or the data load rule.

4. In the Data Load Workbench, verify the values of the data type in the Data Column.

The Data Load Workbench displays the Data Column instead of the Amount Column.

- a. Use the Import from Source feature to import the data from the source system, performs the necessary transformation, such as import, map and validate the data.
- b. Validate the source data.
- c. Export the data to the Oracle Hyperion Planning application.

For more information about how data is loaded in Planning, see the Outline Load Utility in *Oracle Hyperion Planning Administrator's Guide*.

Setting the All Data Types Load Method

You can select the method to use when loading metadata and data for Account, Period, Year, Scenario, Version, Currency, Entity, user-defined dimensions, attributes, UDAs, exchange rates, Smart Lists, and planning unit hierarchies.



Note:

To load numeric data, use the **Numeric Data Only** load method.

1. Select the **Setup** tab, and then under **Register**, select **Target Application**.
2. In **Target Application**, in the **Target Application** summary grid, click **Add**, and then select either **Local** or **Cloud**.

Available options are **Cloud** (for a Cloud deployment) or **Local** (for an on-premise deployment).

3. In **Target Application** under **Application Options**, from the **Load Method** dropdown, select **all data types with security**.

Dimension Details		Application Options	
Property Name	Value	Select	
Load Method	All data types with security		
Batch Size	10000		
Drill Region	No		
Purge Data File	No		
Date Format	MM-DD-YYYY		
Data Dimension for Auto-Increment Line Item			
Driver Dimension for Auto-Increment Line Item			
Member name may contain comma	Yes		

Available load method descriptions:

- All data types with auto-increment of line item.
This method requires that you define the data and driver dimensions, and unique identifiers for the Oracle Enterprise Performance Management Cloud application. You define data load and driver dimension in the Data Dimension for Auto-Increment Line Item field and Driver Dimension for Auto-Increment Line Item field on the Target Options page in the Data Load Rule.

For detailed information on this option, see [Loading Incremental Data using the LINEITEM Flag to an EPM Application](#).

- **Numeric**—Loads numeric data only. Oracle Hyperion Planning data security is not enforced in this method.
 - **all data types with security**— Loads Numeric, Text, Smartlist, Date data types. If the Planning administrator loads data, Planning data security *is not* enforced. If a Planning non-administrator user loads data, then Planning data security is enforced. A Planning non administrator user can load only 500,000 cells of data.
4. From the **Date Format** drop-down, specify the format of the date data:
 - DD-MM-YYYY
 - MM-DD-YYYY
 - YYYY-MM-DD
 5. Click **Save**.

Setting the All Data Types Load Method

There are two export load methods in Oracle Hyperion Financial Data Quality Management, Enterprise Edition: Numeric Data Only and All Data Type with Security. The All Data Type with Security loads only to the currency specified in the import.

You can select the method to use when loading metadata and data for Account, Period, Year, Scenario, Version, Currency, Entity, user-defined dimensions, attributes, UDAs, exchange rates, Smart Lists, and planning unit hierarchies.

Note:

To load numeric data, use the **Numeric Data Only** load method.

1. Select the **Setup** tab, and then under **Register**, select **Target Application**.
2. In **Target Application**, in the **Target Application** summary grid, click **Add**, and then select either **Local** or **Cloud**.

Available options are **Cloud** (for a Cloud deployment) or **Local** (for an on-premise deployment).
3. In **Target Application** under **Application Options**, from the **Load Method** drop-down, select **all data types with security**.

Available load method descriptions:
 - **Numeric**—Loads numeric data only. Oracle Hyperion Planning data security is not enforced in this method.
 - **all data types with security**— Loads Numeric, Text, Smartlist, Date data types. If the Planning administrator loads data, Planning data security *is not* enforced. If a Planning non-administrator user loads data, then Planning data security is enforced. A Planning non administrator user can load only 500,000 cells of data.
4. From the **Date Format** drop-down, specify the format of the date data:
 - DD-MM-YYYY

- MM-DD-YYYY
 - YYYY-MM-DD
5. Click **Save**.

Setting the Import Format Data Types

When loading data, specify the type of data for the import format:

- Delimited–Numeric data
 - Fixed–Numeric data
 - Multi-Column–Numeric Data
 - Delimited–All Data Type
 - Fixed–All Data Type
 - Multi Column–All Data Type
1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
 2. In the **Import Format** summary task bar, select **Add**.
In the upper grid of the Import Formats screen, a row is added.
 3. In **Name**, enter a user-defined identifier for the import format.
You cannot modify the value in this field after a mapping has been created for this import format.
 4. In **Source**, select the source system.
 5. In **Target**, select the target system.
 6. In **Description**, enter a description of the import format.
 7. In **Import Format**, in **File Type**, select the type of non-numeric data file.

Options include:

- Fixed - Numeric Data
See [Defining Import Formats for File-Based Mappings](#).
 - Delimited - Numeric Data
See [Defining Import Formats for File-Based Mappings](#).
 - Multi Column - Numeric Data
See [Loading Multi-Column Numeric Data](#).
 - Delimited - All Data Type
See [Setting the Import Format Data Types](#).
 - Fixed - All Data Type
See [Setting the Import Format Data Types](#).
 - Multi Column - All Data Type
8. If the file type is delimited, in the **File Delimiter** field, select the type of delimiter:
 - comma (,)
 - exclamation (!)
 - semi-colon (;)

- colon (:)
 - pipe (|)
 - tab
 - tilde (~)
9. In **Drill URL**, enter the URL used for the drill-through.
 10. In the **Mapping** section, map any dimensions.
 11. Add the import format expressions.
 12. Click **Save**.

Setting the Import Format for Multi-Column Data Types

You can load data from multiple columns using a combination of column expressions and driver expressions as shown below:

- A Column expression is used to specify the columns to import. You can import a contiguous set of columns or a non-contiguous set of columns.

You specify contiguous columns by using starting and ending columns. For example, 5,10 indicate columns 5 through 10.

You specify non-contiguous columns by using column1 | column2 | column3. For example, 5|7|10 indicates import columns 5, 7 and 10.

- A driver expression can be used to specify the dimension and target value for each column. The driver dimension is the member into which data is loaded. You can have one driver dimension per load, but multiple members can be defined for the driver dimension. The driver dimension is only applicable to the multi-column type.

The member assignment in the data rule is similar to the multi-period assignment. You can assign the driver member in the header record, import format (member expression), or data rule. If you don't specify a header record or member expression in the import format, you can select the member(s) in the data load rule.

For more information, see [Adding an Import Expression for a Data Driver](#) and [Assigning Driver Dimension Members](#).

- For the multi-column type, you can use a header, multi-row header, or no header specified in the import format. These are the different formats:

Note:

In the import format you must have a column definition for the driver dimension defined in the data field. If your driver is "Account," then your import format must include a source column and field, or start and end period for the account dimension. This must be a valid field in the file, or a valid start and end position in the file. This is not referenced by the process, but it must be valid for the process to execute.

- For a file with a header record, use the format Driver=<Dimension Name>; Header=<Row Number>; Column=<Column Numbers>.

For example, when the import format definition
Driver=Account;HeaderRow=1;Column=2,4 is applied to the following sample data file:

```
Entity,ACCT1,ACCT2,ACCT3
Entity01,100,200,300
```

This tells the system that row 1 is the header, and data starts in row 2. In column 2, the entity is the first value, and then the next three columns are the values for ACCT1, ACCT2 and ACCT3.

- For a file with multiple row headers (driver members don't line up with the data column), you can use a modified header expression. For example, when you export data from Essbase as in the following data file, the data column header is a new row and does not line up data.

```
"Period","Consolidation","Data Source","Currency","Intercompany","Entity","Movement","Multi-GAAP","Product","Scenario","Years","View","Account"
"FCCS_Sales","FCCS_Cost of Sales"
"Jan","FCCS_Entity Input","FCCS_Data Input","Entity Currency","FCCS_No Intercompany","01","FCCS_No Movement","FCCS_Local GAAP","P_110","Actual","FY15","FCCS_Periodic",
3108763.22,2405325.62
"Jan","FCCS_Entity Input","FCCS_Data Input","Parent Currency","FCCS_No Intercompany","01","FCCS_No Movement","FCCS_Local GAAP","P_110","Actual","FY15","FCCS_Periodic",
3108763.22,2405325.62
```

With a multi row header, you can instruct the system as to the header row that contains the driver information. When the header row is specified as Header=2,1, this means that the header starts at row 2, and the driver members start at column 1.

In another example, say your second header is A,B,C,D and columns are 10 to 13 for these values. If you set column expression to 10|12,13, then the B member and its values (at column 11) are skipped.

- To load multiple columns without a header record in the data file, use the import format definition Driver = <Dimension Name>; Member = <List of Members>; Column=<Column Numbers>. You may want to use this method when you want to skip a source column in the source record.

For example, when the import format definition Driver=Account;member=ACCT1,ACCT2, ACCT3;Column=2,4; is applied to the following data file:

```
Entity01,100,200,300
```

you tell the system to include entity as the first value, and then for the next three columns to use driver dimension members values from ACCOUNT; ACCT1, ACCT2 and ACCT3.

- For data source application types, you assign the driver dimension, but the system assigns row 1 as the header. You can load multiple columns by selecting the columns from the Add Import Format Mapping Expression screen.

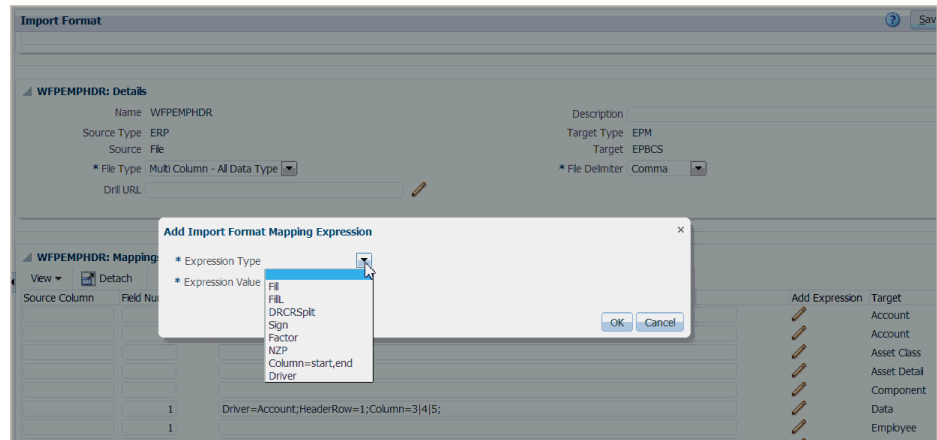
Adding an Import Expression for a Data Driver

To add an import expression for non-numeric data:

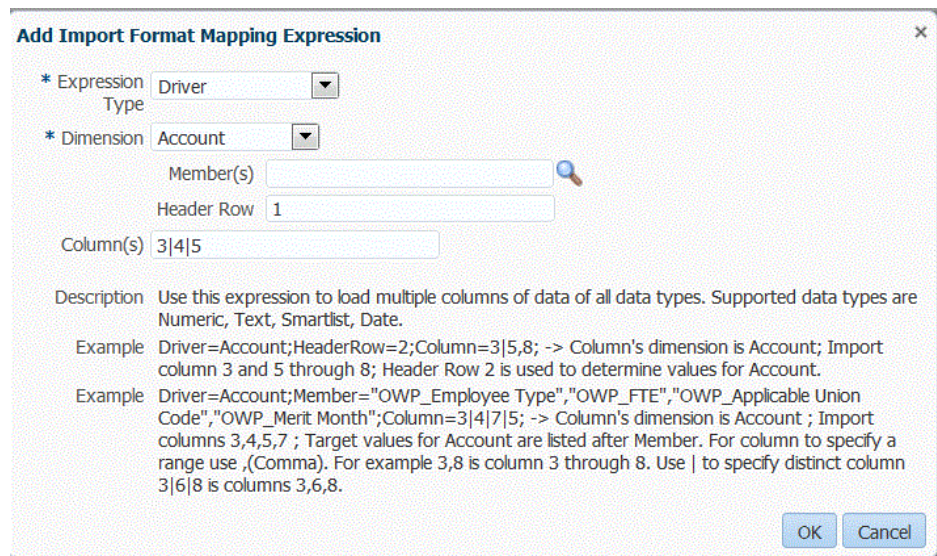
- On the **Setup** tab, under **Integration Setup**, select **Import Format**.
- In **Import Format**, from the **Import Format Mapping** grid, select the data source column.
- In **Expression**, specify the import expression.
- Optional:** You can also specify the expression type and value in the **Add Expression** field.

- Click .

- b. From the **Expression Type** drop-down, select **Driver**.

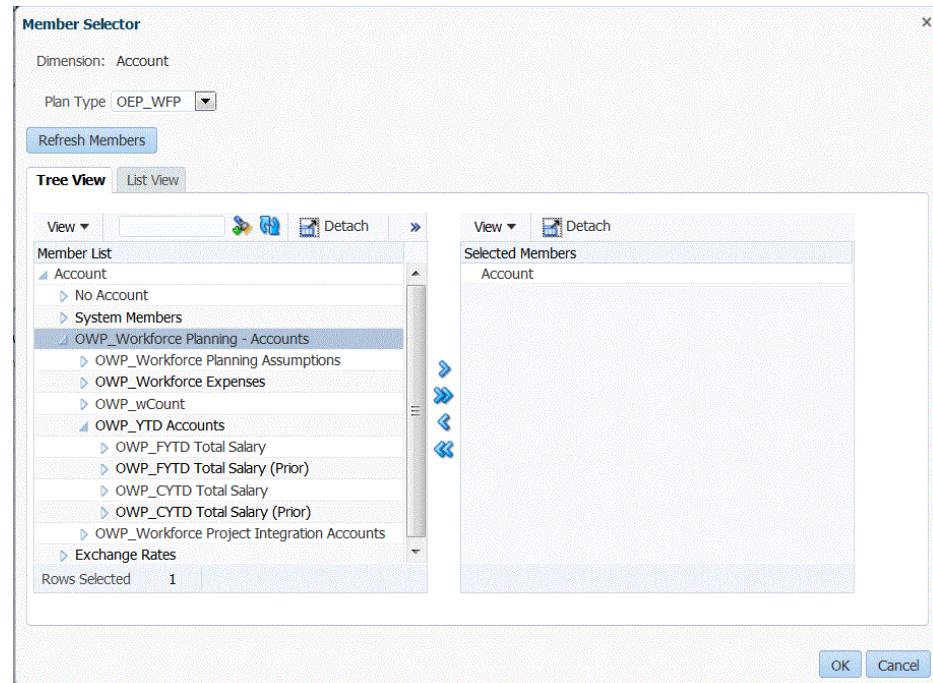


- c. In **Add Import Format Mapping Expression** when entering a driver, enter the values for the expression and click **OK**.



In **Dimension**, select the dimension to which the expression applies.

In **Member(s)**, select the dimension members. You can also browse and select members on the Member Selector screen.

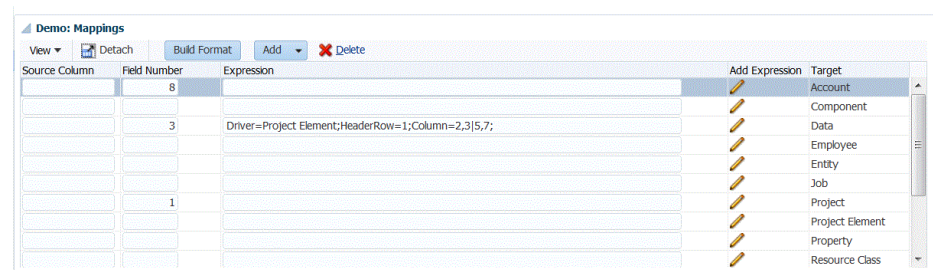


In **Header row**, select the header row of the file for the expression.

In **Column(s)**, specify the data columns in the expression. To use a range of DATA columns, specify columns using a comma (.). To use non-contiguous DATA columns, specify columns using the pipe (|) delimiter.

- d. Click **OK**.

In the following example, the "Project Element" is the driver member of the first header row, and includes contiguous rows "2,3", and non-contiguous rows "5,7".



Assigning Driver Dimension Members

A driver dimension member can be assigned in the data rule (as well as the header record or import format (member expression)). This tab is only displayed in the data load rule if the members are not specified in the import format, or if the header row is not specified in the import format.

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. In the **Data Load** summary area, click **Add** to add a new date load rule, or change the location name from the POV bar, and select the data load rule.

For information on adding a data load rule, see [Defining Data Load Rule Details for a File-Based Source System](#).

3. Select the **Target Member** tab.

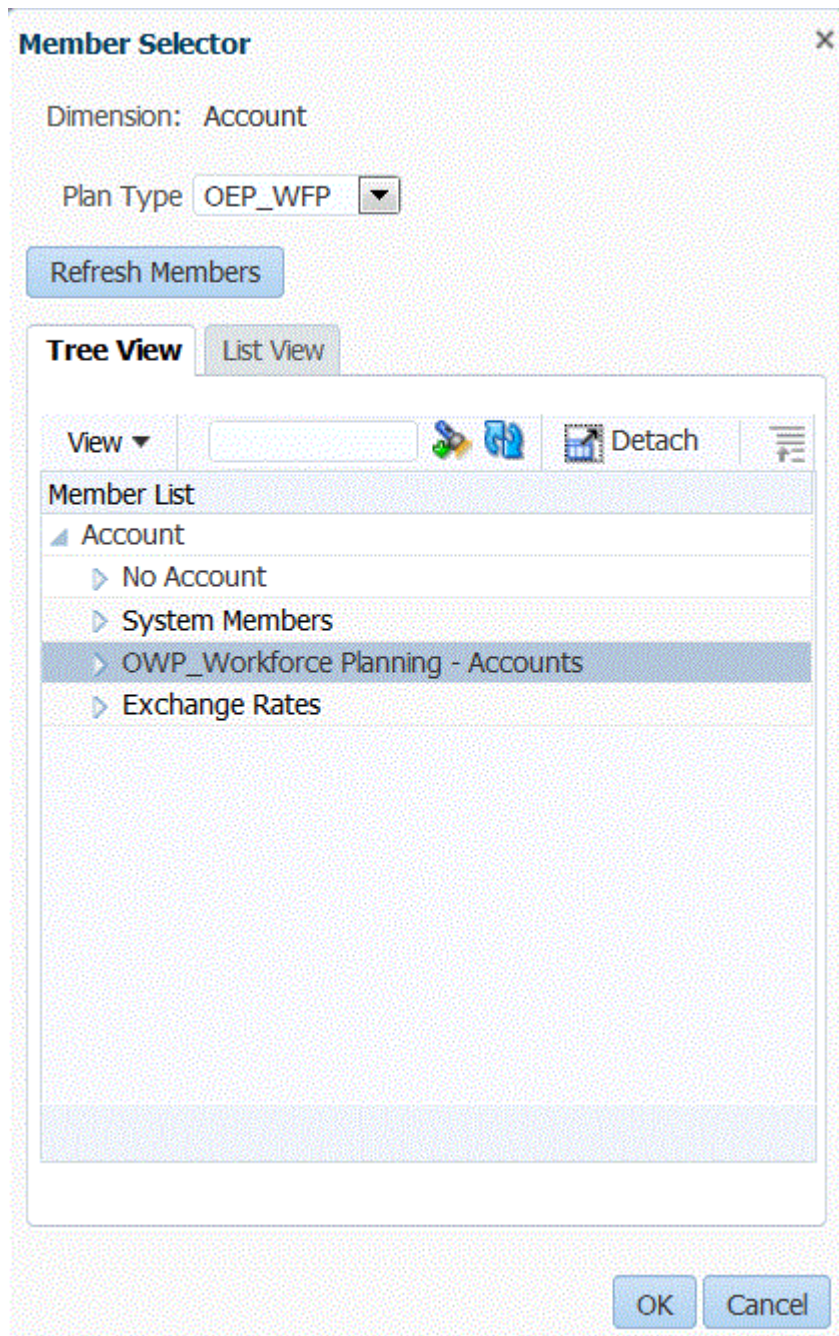
You see this tab only when the driver dimension members are not defined in the import format.

The columns to which you can select driver dimension members are available for edit.

The screenshot shows the 'Details' tab of a configuration window. At the top, there are fields for 'Name' (DriverMembers), 'Category' (OEP_Plan), 'Import Format', 'Description', and 'Target Plan Type' (Plan1). Below these are three tabs: 'Source Options', 'Target Members' (which is selected), and 'Custom Options'. The 'Target Members' tab contains a table with columns 'Target Column', 'Value', and 'Select'. The table has two data rows: 'Column10' and 'Column12'. The 'Value' column for 'Column10' is highlighted, and the 'Select' column for 'Column12' has a search icon.

Target Column	Value	Select
Column10		
Column12		

4. In the **Value** field, enter the name of the driver dimension member to use in the header record or member expression.
5. **Optional:** To search on driver dimension members, click the Search button and navigate to the driver dimension on the Member Selection screen.



6. Click **Save**.

Loading Incremental Data using the LINEITEM Flag to an EPM Application

You can include line item detail using a LINEITEM flag in the data load file to perform incremental data loads for a child of the data load dimension based on unique driver dimension identifiers to an EPM application. This load method specifies that data should be overwritten if a row with the specified unique identifiers already exists on the form. If the row does not exist, data is entered as long as enough child members exist under the data load dimension parent member.

For example, you can load employee earnings detail from the following sample source data file to a target EPM application.

Emp, Job, Pay Type, Amount
 "Stark, Rob", Accountant, Bonus_Pay, 20000
 "Molinari, Sara", Sales Manager, Bonus_Pay, 22000
 "Matthew, Peter", Sales Associate, Bonus_Pay, 5000

The target Oracle Hyperion Planning application is shown below:

Employee Earnings ⓘ							
Scenario	Version	Currency	Entity	Project	Years	Employee	Job
OEP_Plan	OEP_Working	USD	Sales US	No Project	FY16	Stark, Rob	Accountant
		Jan					
		No Property	OWP_Value				
Earning1	Bonus Pay	25,000					
Earning2							
Earning3							
Earning4							
Earning5							

When using the LINEITEM syntax, the data file may contain records having identical dimensions except driver member values.

In the following data file, records have the same dimensions but differ on the value of the acct_date column (a driver member). This requires you to identify driver member(s) which make the data record unique (that is, the acct_date column for the example).

```
Entity, Employee, Version, asl_EmployeeType, acct_date, acct_text, SSTax_Rate1
<LINEITEM("ParentMember")>, No Employee, Baseline, Regular, 1-1-2001, Text1, 0.4
<LINEITEM("ParentMember")>, No Employee, Baseline, Regular, 1-1-2002, Text2, 0.4
<LINEITEM("ParentMember")>, No Employee, Baseline, Regular, 1-1-2003, Text3, 0.5
```

To support the above use case, create a LOOKUP dimension and map driver member column to it in the Import Format option. The name of the dimension must start with LineItemKey. For example, create a LOOKUP dimension named LineItemKey and assign any Data Column Name (such as UD8). In the Import Format option, map LineItemKey dimension to 5th column (acct_date) in the data file and use the LIKE (* to *) data mapping. You may also use other type of data mappings to populate the look up dimension. If needed, create more LOOKUP dimensions to uniquely identify data records. The rest of the setup is same.

To use this feature, you need to perform steps both in Planning and Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

1. Launch Planning.
2. From the **Data Load Settings** screen, select the **Data Load Dimension** and **Driver Dimension**.

In Planning, **Earning1** and **Earning2**, are members of the **Account** dimensions. The various Earnings Types are loaded to the **No Property** member of the **Property** dimension, and the **Earning** value is loaded to the **OWP_Value** of the **Property** dimension.

The first screenshot shows the 'Data Load Settings' section with 'Data Load Dimension' set to 'Account' and 'Driver Dimension' set to 'Property'. The second screenshot shows the 'Advanced Settings' section with a table for 'Data Load Dimension Parent' and 'Driver Dimension Unique Identifiers'.

Data Load Dimension Parent	Driver Dimension Unique Identifiers
OWP_Total Earnings	"No Property"

For more information about the Data Load Settings screen, see the *Oracle Hyperion Planning Administrator's Guide*.

3. Launch FDMEE, then select **Setup**, and then select **Import Format**.
4. From the **Import Format Mapping** grid, select the data source column.
5. In **Expression**, add an import expression for the data driver.

For example, add the import format expression: **Driver=Property;member="No Property";"OWP_value";Column=3,4**.

The first screenshot shows the 'WFEMPCOMP: Details' section with fields for Name, Source, File Type, Drill URL, Description, Target, and File Delimiter. The second screenshot shows the 'WFEMPCOMP: Mappings' section with a table for Source Column, Field Number, Expression, Add Expression, and Target.

Source Column	Field Number	Expression	Add Expression	Target
	3			Account
				Asset Class
				Asset Detail
				Component
Pay Type	1	Driver=Property;member="No Property";"OWP_value";Column=3,4		Data
Emp	1			Employee
				Entity
Job	2			Job
				Market

For more information about adding drivers FDMEE, see [Adding an Import Expression for a Data Driver](#) and [Assigning Driver Dimension Members](#).

6. From **Workflow**, select **Data Load Mapping**.

In Data Load Mapping, you identify how source dimensionality translates to the target dimensionality. As shown below for a "Like" mapping, the **Earning** source value (represented by the asterisk) is loaded to **OWP_Total Earnings** of the **Account** dimension.

The screenshot shows the 'Data Load Mapping' section with tabs for All Mappings, Explicit, Between, In, Multi Dimension, and Like. The 'Like' tab is selected, showing a table for Source Value, Target Value, Script, Change Sign, and Rule Name.

Source Value	Target Value	Script	Change Sign	Rule Name
*	<LINEITEM("OWP_Total Earnings")>			L999

7. From **Workflow**, select **Data Load Rule**.
8. Select the **Target Options** tab.
9. From the **Load Method** drop-down, select **All data types with auto-increment of line item**.

10. In **Data Dimension for Auto-Increment Line Item**, select the data dimension that matches the data dimension you specified in Planning.

In this example, the data dimension is **Account**.

11. In **Driver Dimension for Auto-Increment Line Item**, select the driver dimension that matches the driver dimension you specified in Planning.

In this example, the driver dimension is **Property**.

Dimension Details			Application Options	
View ▾  Detach				
Property Name	Value			Select
Load Method	All data types with auto-increment of line item			
Batch Size	10000			
Drill Region	Yes			
Purge Data File	Yes			
Date Format	MM-DD-YYYY			
Data Dimension for Auto-Increment Line Item	Account			
Driver Dimension for Auto-Increment Line Item	Property			
Member name may contain comma	Yes			

Loading Multi-Column Numeric Data

Oracle Hyperion Financial Data Quality Management, Enterprise Edition enables you to load data for multiple dimension members of a selected dimension in a single row of data. The definition of the members to load can be included in a header record in the load file, or in the import format definition.

- text data file with multi-columns of numeric data with no headers to contiguous periods by executing a data load rule with start and end periods.
- text data file with multiple columns of numeric data to a period or any other dimension as a column header by specifying the:
 - column header in the data file
 - column header member list in the import format
 - column header member in the data rule
- Excel data file with multiple columns of numeric data to a period as a column header. The Excel file may or may not contain a header.

To load multi-column numeric data:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. In the **Import Format** summary task bar, select **Add**.
In the upper grid of the Import Formats screen, a row is added.
3. In **Name**, enter a user-defined identifier for the import format.
You cannot modify the value in this field after a mapping has been created for this import format.
4. In **Description**, enter a description of the import format.
5. In **Source**, select **File** for the source.
6. From **File Type** drop-down, select **Multi Column - Numeric Data** as the format of the file.
7. From the **File Delimiter** drop-down, select a type of delimiter.

Available delimiter symbols:

- comma (,)
- exclamation (!)
- semicolon (;)
- colon (:)
- pipe (|)
- tab
- tilde (~)

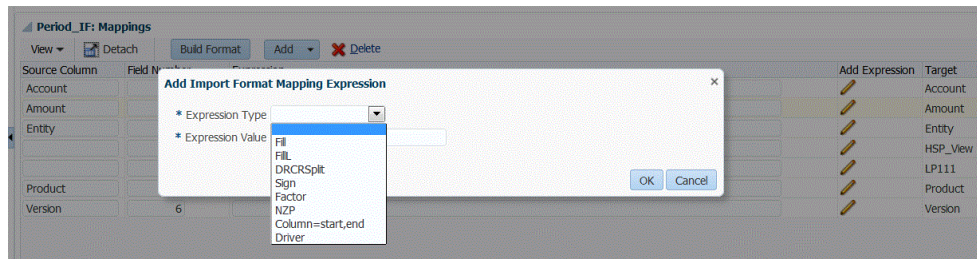
8. In **Target**, select **EPM** and select any EPM application as a target.

9. **Optional:** In **Drill URL**, enter the URL used for the drill-through.

10. In the **Mapping** section, select the **Amount** dimensions

and click .

11. From the **Expression Type** drop-down, select **Column=start,end**.

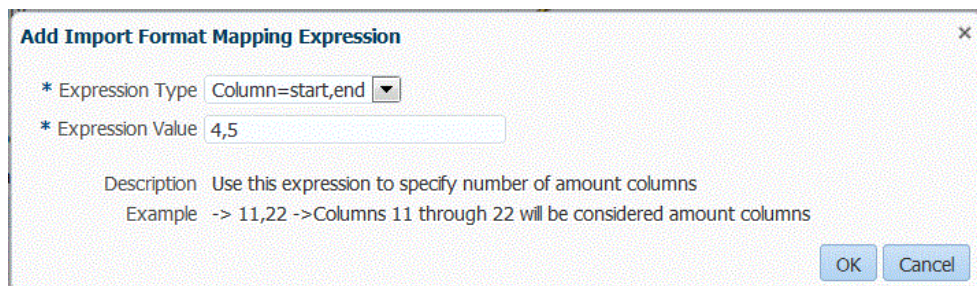


12. In **Expression value**, specify the columns to import.

You can import a contiguous set of columns or a non-contiguous set of columns. To use a range of Amount (data) columns, specify columns using a comma (,). To use non-contiguous amount columns, specify columns using the pipe (|) delimiter.

You specify contiguous columns by using starting and ending columns. For example, 5,10 indicates columns 5 through 10.

You specify non-contiguous columns by using column1 | column2 | column3. For example, 5|7|10 indicates import columns 5, 7, and 10.



13. **Optional:** Specify any drivers and header rows of the file for the expression.

14. Click **OK**.

Acc_1: Mappings

View Detach Build Format Add Delete

Source Column	Field Number	Expression	Add Expression	Target
Account	1			Account
Amount	1	Driver=Account;HeaderRow=1;Column=1,3		Amount
Custom1	6			Custom1
Entity	4			Entity
Version	5			Version

15. Click **Save**.

To load a text data file with multiple columns of numeric data to a period:

1. Complete steps 1-12 in the [To load multi-column numeric data](#).
2. From the **Expression Type** drop-down, select **Driver**.
3. On the **Add Import Format Mapping Expression**, in **Dimension**, leave the default driver dimension **Period**.
4. In **Period(s)**, select the **period driver dimension member** to load and click **OK**.

Specify the period using quotes. For example, you might enter "**Dec-9**".

If you do not specify a period driver member dimension on the Add Import Format Mapping Expression, you can specify period members in the data load rule. See steps 5-11.

Add Import Format Mapping Expression

* Expression Type: Driver

* Dimension: Period

Period(s): "Dec-9"

Header Row:

Column(s): 4,5

Description: Use this expression to load multiple columns of data of all data types. Supported data types are Numeric, Text, Smartlist, Date.

Example: Driver=Account;HeaderRow=2;Column=3|5,8; -> Column's dimension is Account; Import column 3 and 5 through 8; Header Row 2 is used to determine values for Account.

Example: Driver=Account;Member="OWP_Employee Type","OWP_FTE","OWP_Applicable Union Code","OWP_Merit Month";Column=3|4|7|5; -> Column's dimension is Account ; Import columns 3,4,5,7 ; Target values for Account are listed after Member. For column to specify a range use ,(Comma). For example 3,8 is column 3 through 8. Use | to specify distinct column 3|6|8 is columns 3,6,8.

OK Cancel

5. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
6. On the **Data Load Rule** screen, select the POV to use for the data load rule.
7. Add or select the data load rule to use for the multi-column numeric data load.
8. In **Import Format**, select the import format set up for the multi-column numeric load.
9. **Optional:** From the **Source Options** tab, specify any source options.
10. Select the **Column Headers** tab, and specify the start date and end date of the numeric columns.

You are prompted to add the start and end dates on the Column Headers tab when:

- a text data file has no header in the header record of the data file, in the import format, or data rule.
- you are using an Excel file in all cases. If header information is specified in the Excel file, only periods that fall within the start and end period range are processed.

Source Options

Column Headers

Target Options

Custom Options

View ▾

Detach

Column Number	Value	
4	Dec-24	
5	Jun-24	

11. Save and execute the data load rule.

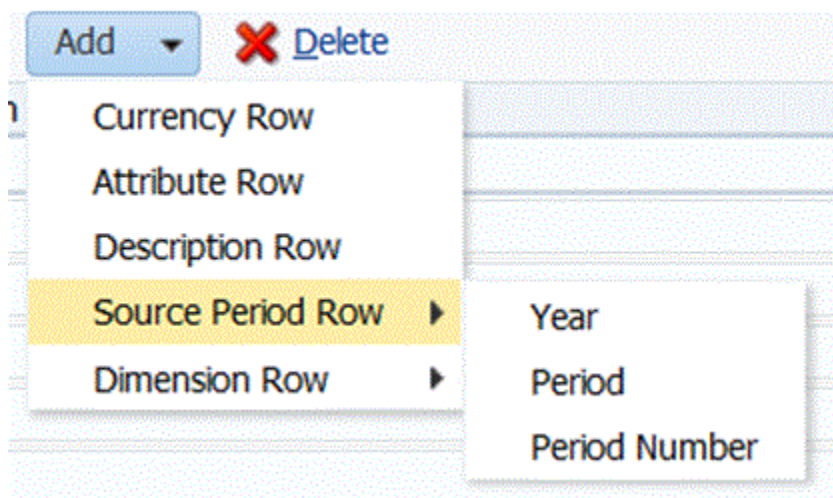
Loading Periods As a Column from the Data File

“Period” dimensions are supported as columns in a data file. If you have data for multiple periods in a single file, then you can include the year and period on each row of the data file that gets loaded to the target application.

You load a period as a column from a data file by defining the load definition through the import format, and the data load rule.

To include a period as a column from the data file:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. In the **Import Format** summary task bar, select the Oracle Financial Consolidation and Close Cloud consolidation application or file.
Typically this is a single file with multiple periods, such as Year and Period.
3. From the **Import Format Detail Mapping** grid, select the source column and click **Add**.



4. Select the period rows to include:
 - a. To map to a "Year" row, click **Source Period Row**, and then select **Year** to map to the **Year** dimension in the target application.

- b. In **Source Column**, select the dimension to map to the **Year** row.
 - c. In **Field Number**, enter the **field number** from the file to import (defaults to the field number from the file when text is selected.)
 - d. In **Expression**, specify the expression to apply to the **Year** row.
 - e. To map to a period row, click **Add**, select **Source Period Row**, and then select **Period**.
 - f. In **Field Number**, enter the **field number** from the file to import (defaults to the field number from the file when text is selected.)
 - g. In **Expression**, specify the expression to apply to the **Period** row.
 - h. To map to a period number row, click **Add**, select **Source Period Row**, and then select **Period Number**.
 - i. In **Field Number**, enter the **Field number** from the file to import (defaults to the field number from the file when text is selected.)
 - j. In **Expression**, specify the expression to apply to the **Period Number** row.
5. Click **Save**.
 6. Specify the parameters of the data load rule, and then execute it.
See [Defining Data Load Rules to Extract Data](#).

Write-Back Mappings

When you are setting up an integration to pull from an EPM application and write back to Oracle E-Business Suite or PeopleSoft, consider the following:

- Before you create a data load rule to write back, create the write-back mappings. Write-back mappings occur at the member level. (For example, loading data from an Oracle Hyperion Planning application to your Enterprise Resource Planning (ERP) source system.)
- You create write-back mappings to replace outgoing dimension members with source segment members. More specifically, during budget write-back, the write-back mapping is referred to when replacing outgoing dimension members with segment values.

The following interface tables require “write” security privileges for the data load to write-back process:

E-Business Suite

- GL_INTERFACE
- GL_INTERFACE_CONTROL
- GL_BUDGET_INTERFACE

Standard PeopleSoft—PS_HPYPB_ACCT_LN

PeopleSoft Commitment Control

- PS_HYP_KK_BD_HDR
- PS_HYP_KK_BD_LN

To create write-back mappings:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.

2. In **Import Format**, if necessary, select the **Write Back Mapping** tab.
3. In **Source Dimension**, select the source dimension to map to the source dimension.
4. **Optional:** To have Oracle Hyperion Financial Data Quality Management, Enterprise Edition create the write back mapping automatically, click **Auto Create**.
When prompted to create the write back mapping, click **OK**.
5. Click **Save**.

Defining Import Formats for Data Synchronization Between EPM Applications

The import format determines which fields (columns) to store and push from the Oracle Hyperion Planning application to the Oracle Financial Consolidation and Close Cloud target application dimensions.

To add an import format for data synchronization between EPM applications:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. In the **Import Format** summary task bar, select **Add**.
In the upper grid of the Import Formats screen, a row is added.
3. In **Name**, enter a user-defined identifier for the import format.
The value in this field cannot be modified after a mapping is created for this import format.
4. In **Description**, enter a description of the import format.
5. In **Source**, select the source system.
For an EPM source system, select the EPM source application or file from which to move data.
For an Enterprise Resource Planning (ERP) source system, select the ERP source application (for example Oracle E-Business Suite, Peoplesoft Financials, Fusion) from which to move data.
6. In **Target**, select the target system.
For an EPM target system, select the EPM target application to which to move data.
For an Enterprise Resource Planning (ERP) target system, select the ERP target application to which to move data.
When you use an E-Business Suite and Fusion target system, you must enter the Chart of Accounts. The Accounting Entity is captured at the Location or Data Rule level.
7. From **Drill URL**, specify the drill information.
See [Adding Drill Through URLs](#).
8. Define the import mappings from the EPM or Enterprise Resource Planning (ERP) application.

To add an import mapping for an EPM source application to an EPM target application.

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. In **Import Format**, from the **Import Format** summary grid, select an EPM source application.
3. In the **Import Format Detail** grid, and then in **Source Column**, select the dimension to map.
4. **Optional:** Add other dimensions as needed by selecting a dimension type from the **Add** drop-down.
Available options:
 - Currency Row
 - Attribute Row
 - Description Row
 - Dimension Row
5. In the **Expression** field, enter an expression or import script to the import format.
When entering a constant, you must still enter a starting position and length. Use a start position of "1 " and a length of "1."
See [Adding Import Expressions](#).
6. Click **Save**.

Defining Import Formats for Data Synchronization Between Enterprise Resource Planning (ERP) and EPM Applications

Data synchronization also enables Oracle Hyperion Financial Data Quality Management, Enterprise Edition to map the Oracle Enterprise Performance Management System application dimension to the Enterprise Resource Planning (ERP) Segment/Chartfield.

To create Enterprise Resource Planning (ERP) to EPM mappings:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. In **Import Format**, from the **Import Format** summary grid, select an Enterprise Resource Planning (ERP) source application.
Depending on your selection in this field, the Source field shows only the registered Enterprise Resource Planning (ERP) sources and file for an ERP source type; or registered EPM applications.
For example, the Source field may be an Oracle Hyperion Financial Management or Oracle Hyperion Planning for an EPM source type.
3. In the **Import Format Detail** grid, select the dimension to map from the **Source Column** drop-down.
4. **Optional:** Add other dimensions as needed by selecting a dimension type from the **Add** drop-down.
Available options include:
 - Currency Row
 - Attribute Row
 - Description Row

- Dimension Row
5. In the **Expression** field, enter an expression or import script to the import format.
When entering a constant, you must still enter a starting position and length. Use a start position of "1 " and a length of "1."
See [Adding Import Expressions](#).

6. Click **Save**.

To create EPM to Enterprise Resource Planning (ERP) (Write-back) mapping:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. In **Import Format**, from the **Import Format** summary grid, select an EPM source application.
3. Select the **Segment**.
4. Choose the type of mapping by selecting either the Explicit tab, Between tab, Multi-Dimension, or Like tab.
 - **Explicit**—The source value is matched exactly and replaced with the target value. For example, the source value, "ABC" is replaced with the target value, "123." "Explicit" write-back mappings are created the same for data load and data write-back rules. See [Creating Mappings Using the Explicit Method](#).
 - **Between**—The range of source values is replaced with a single target value. For example, a range from "001" to "010" is replaced as one value: "999." "Between" write-back mappings are created the same for data load and data write-back rules. See [Creating Mappings Using the Between Method](#).
 - **In**—In mappings enable a list of non-sequential source accounts to be mapped to one target account. In this case, multiple accounts are mapped to one account within one rule, eliminating the need to create multiple rules (as is required for an Explicit map).
 - **Like**—The string in the source value is matched and replaced with the target value. For example, the source value, "Department" is replaced with the target value, 'Cost Center A. See [Creating Mappings Using the Like Method](#).

Write-back mappings provide a means to remove or strip characters that were added during the data load process. "Like" write back mappings are created similar to, but reverse from the data load.

- **Multi-Dimension**—Define member mapping based on multiple source column values.

 Tip:

You can click Refresh Values to refresh the list of segment or chartfield values that appear in the drop-down list from the source system. This is especially helpful when creating "Explicit," "Between," "Like", and "Multi-Dimension" mappings for data write-back data loads.

Defining the Write-Back Mappings (E-Business Suite and PeopleSoft only)

When you define an import format, you can also define the import format mappings from the EPM application for the data write-back rule.

To define the write-back import format mappings:

1. In **Import Format**, select the **Import Name**.
2. In **Source** select the **Planning** or **Essbase** as the source system.
3. In **Target** , select **EBS** or **PeopleSoft** as a target system.
4. Select the **Accounting Entity**.
Select **Ledger** for Oracle E-Business Suite.
Select **Business Unit** for PeopleSoft.
5. Click **Save**.

Defining Locations

A location is associated with one source system, but you can import data from multiple ledgers from that system. Each location is assigned an import format enabling you to use the same import format for more than one target application where the dimensionality of the target applications is the same.

The Location feature also enables you to specify free form text or a value using the integration option feature. Text or values entered for a location can be used with your Oracle Hyperion Financial Data Quality Management, Enterprise Edition scripts.

Note:

You can create duplicate locations with the same source system and application combination.

To create, edit, and delete import locations:

1. On the **Setup** tab, under **Integration Setup**, select **Location**.
2. In **Location**, click **Add**.
3. From **Location Details**, in **Name**, enter the location name.
4. From **Import Format**, enter the import format.

The Import Format describes the source file structure, and it is executed during the source file import step. FDMEE supports file-based imports for those users who do not have direct connections to their source data, but do have source data residing in a text file. A corresponding import format must exist before it can be used with a location.

Additionally:

- The source system is automatically populated based on the import format.

- The Source field displays the source application based on the import format.

You can also click  and select an import format.

5. In **Accounting Entity**, specify the source system accounting entities (business units or ledgers) to use for the location.

For Fusion and E-Business Suite source systems, the Accounting Entity is the ledger. For PeopleSoft source systems, the Accounting Entity is the business unit.

If the accounting entity is selected here, then in the Data Load Rules, accounting entity populates automatically.

If the accounting entity is not specified here, you can specify the accounting entity in data rules. Doing so enables you to load data from multiple ledgers to business units from one location.

You can use locations with multiple import formats. For example, you can define the import format for the first location, Ledger 1. Then, define another import format for Ledger 2. In this case, you would create multiple locations with the same import format. You can also define multiple locations for target applications that are the same. In this case, you can define multiple locations for each business unit or ledger and reuse the import format.

6. In **Target Application**, specify the target application associated with this location.

You can also click  to search for a target application.

7. **Optional:** Enter or click  to select the **Accounting Entity**. (For E-Business Suite, select the ledger. For PeopleSoft, select the business unit.)
8. In **Functional Currency**, specify the currency of the location.

 Note:

When you create a location using a PeopleSoft Human Capital Management source system, the Functional Currency column displays "NA." Unlike PeopleSoft Enterprise Financial Management source systems, when FDMEE human resources data loads occur, there is no break out of amount by functional, entered, or translated currencies.

9. In **Parent Location**, enter the parent assigned to the location.

Parent mapping are used to share mappings with other locations. Enter mappings at the parent location, and the related locations can use the same mappings. Multiple locations can share a parent. This feature is useful when multiple locations use one chart of accounts. Changes to a child or parent mapping table apply to all child and parent locations.

10. In **Data Value**, specify the extra dimension that is used only for integration with multiple dimension target systems.

This dimension is associated with a data load location. In Oracle Hyperion Financial Management, the data value is the value dimension. When FDMEE creates the load file, the dimension value is entered for every data line that is loaded by the location. For example, the Data Value dimension is associated with the Value dimension in Financial Management. By default, if no value is entered in

this field when integrating with Financial Management, the Data Value <EntityCurrency> is the default value.

When Search is selected, FDMEE connects to the Financial Management to get a list of valid data values. FDMEE takes the values from Financial Management and adds rows created by FDMEE that are a concatenation of the original value and "Adjustment Data Values." FDMEE uses these newly created rows to manage journal loading to Financial Management.

The rows that FDMEE creates in the Data Value selection screen are:

- [Contribution Adjs];[Contribution Adjs]
- [Contribution Adjs];[Parent Adjs]
- [Contribution Adjs];<Entity Curr Adjs>
- [Contribution Adjs];<Parent Curr Adjs>
- [Parent Adjs];[Contribution Adjs]
- [Parent Adjs];[Parent Adjs]
- [Parent Adjs];<Entity Curr Adjs>
- [Parent Adjs];<Parent Curr Adjs>
- <Entity Curr Adjs>;[Contribution Adjs]
- <Entity Curr Adjs>;[Parent Adjs]
- <Entity Curr Adjs>;<Entity Curr Adjs>
- <Entity Curr Adjs>;<Parent Curr Adjs>

- 11. Optional:** In **Logic Account Group**, specify the logic account group to assign to the location.

A logic group contains one or more logic accounts that are generated after a source file is loaded. Logic accounts are calculated accounts that are derived from the source data.

The list of values for a logic group is automatically filtered based on the Target Application under which it was created.

- 12. Optional:** In **Check Entity Group**, specify the check entity group to assign to the location.

When a check entities group is assigned to the location, the check report runs for all entities that are defined in the group. If no check entities group is assigned to the location, the check report runs for each entity that was loaded to the target system. FDMEE check reports retrieve values directly from the target system, FDMEE source data, or FDMEE converted data.

The list of values for a check entity group is automatically filtered based on the Target Application under which it was created.

- 13. Optional:** In **Check Rule Group**, specify the check rule group to assign to the location.

System administrators use check rules to enforce data integrity. A set of check rules is created within a check rule group, and the check rule group is assigned to a location. Then, after data is loaded to the target system, a check report is generated.

The list of values for a check rule group is automatically filtered based on the Target Application under which it was created.

14. Click **Save**.

15. **Optional:** Perform these tasks:

- To edit an existing location, select the location to modify, and then make changes as necessary. Then, click **Save**.
- To delete a location, click **Delete**.

When a location is deleted, the location is removed from all other FDMEE screens, such as Metadata, and Data Load.

 Tip:

To filter by the location name, ensure that the filter row is displayed above the column headers. (Click  to toggle the filter row.) Then, enter the text to filter.

To specify free form text or values for use with scripts:

1. On the **Setup** tab, under **Integration Setup**, select **Location**.
2. In the **Location Detail**, click the **Integration Option** tab.
3. From **Integration Option**, in **Integration Option 1-4**, specify the free form text or value, and then click **OK**.

The information is accessible from the Integration Option fields in the Location table.

Defining Period Mappings

You have the flexibility to use various kinds of calendars (for example, monthly, weekly, or daily) based on your business and statutory requirements. In your EPM system, you can also use different calendars, based on your application requirements (for example, different levels of periods). Because Oracle Hyperion Financial Data Quality Management, Enterprise Edition extracts the Enterprise Resource Planning (ERP) source system data to the target EPM application, establish the mapping relationship by defining a period mapping between the source ERP source system periods and the target EPM application periods.

Before you can define data rules, define the period mappings. Period mappings define the mapping between Enterprise Resource Planning (ERP) calendars and the EPM application year or periods. You can define period mappings in three ways:

- **Global Mapping**—You define a global mapping in cases where you do not have many target applications getting data from multiple source systems with different types of source calendars. Use a global mapping to ensure that various periods are accommodated in the individual mapping. As a first step, define a global mapping.
- **Application Mapping**—If you have multiple target applications, getting data from various source systems with complex period types, you can create application mappings in addition to global mappings. When you define an application mapping, you can modify the Target Period Month as necessary.
- **Source Mapping**—Specifies source period mapping for adapter based integrations.

Global Mapping—Sample Monthly Period Mapping

The following table shows how a monthly calendar from a source maps to monthly periods in a target application.

 Note:

You should define global mapping at the most granular level. For example, if you have a monthly calendar and a weekly calendar, define your global mapping at the lowest level of granularity. In this case, the period keys are at the week level and you map weeks to months. You can create application mappings for the higher-level periods.

Table 2-8 Sample Monthly Period Mapping

Period Key	Prior Period Key	Period Name	Target Period Month	Target Period Quarter	Target Period Year	Target Period Day	Year Target
Jan 1 2010	Dec 1 2009	January 1, 2010	Jan	Q1			FY10
Feb 1 2010	Jan 1 2010	February 1, 2010	Feb	Q1			FY10
Mar 1 2010	Feb 1 2010	March 1, 2010	Mar	Q1			FY10
April 1 2010	March 1 2010	April 1, 2010	Apr	Q2			FY10
May 1 2010	April 1 2010	May 1, 2010	May	Q2			FY10

Global Mapping—Sample Weekly Period Mapping

The following table shows how a weekly calendar from an Enterprise Resource Planning (ERP) source system maps to monthly periods in the EPM application.

Table 2-9 Sample Weekly Period Mapping

Period Key	Prior Period Key	Period Name	Target Period Month	Target Period Quarter	Target Period Year	Target Period Day	Year Target
Jan 26 2009	Jan 19 2009	January 26, 2010	Jan	Q1			FY09
Feb 2 2009	Jan 26 2009	February 2, 2010	Feb	Q1			FY09
Feb 9 2009	Feb 2 2009	February 9, 2010	Feb	Q1			FY09
Feb 16 2009	Feb 9 2009	February 16, 2010	Feb	Q1			FY09

Application Mapping—Sample Target Application Sourcing from a Monthly Calendar Source

The following table shows a sample where the target application is sourcing from a monthly calendar. This mapping is performed on the Application Mapping tab.

Table 2-10 Sample Application Mapping—Target Application #1 with a Monthly Calendar Source

Period Key	Target Period Month	Target Period Quarter	Target Period Year	Target Period Day	Year Target
Jan 1 2009	Jan	Q1			FY09
Feb 1 2009	Feb	Q1			FY09
Mar 1 2009	Mar	Q1			FY09

Application Mapping—Sample Target Application #2 Sourcing from a Weekly Calendar Source

The following table shows a sample where the target application is derived from a weekly calendar. This mapping is performed on the Application Mapping tab.

Table 2-11 Sample Application Mapping—Target Application #2 with a Weekly Calendar Source

Period Key	Target Period Month	Target Period Quarter	Target Period Year	Target Period Day	Year Target
Jan 26 2009	Jan	Q1			FY09
Feb 2 2009	Feb	Q1			FY09
Feb 9 2009	Feb	Q1			FY09
Feb 16 2009	Feb	Q1			FY09

**Note:**

To avoid double counting on Income Statement accounts, be sure not to define a mapping where the adjustment period of one year goes into the period of the next fiscal year.

Adjustment Period Mapping—Mapping the Period Key to the Adjustment Period**Note:**

If YTD is selected as the Enterprise Resource Planning (ERP) source, then the adjustment period becomes the ending balance (replaces the period 12). If PTD, then the adjustment period gets added to period 12.

Table 2-12 Sample Adjustment Period Mapping—Mapping the period to the adjustment period

Period Key	Calendar	Adjustment Period	Description
Dec-2003	Accounting 13	13-03	Adj Period for 2003
Dec-2004	Accounting 13	13-04	Adj Period for 2004
Dec-2005	Accounting 13	13-05	Adj Period for 2005

Table 2-12 (Cont.) Sample Adjustment Period Mapping—Mapping the period to the adjustment period

Period Key	Calendar	Adjustment Period	Description
Dec-2007	Accounting 13	13-07	Adj Period for 2007

 Note:

If the source is PeopleSoft General Ledger, set the adjustment period mapping with the related accounting year.

Global Mappings

You can define one global mapping to map various periods to the individual mapping.

To define a global mapping:

1. On the **Setup** tab, under **Integration Setup**, select **Period Mapping**.
2. Select the **Global Mapping** tab.
3. Click **Add**.
4. Select the **Period Key**.
5. Select the **Prior Period Key**.
6. Enter the following:
 - a. Period Name; for example, August 2005.
 - b. Target Period Month; for example, August.
 - c. Target Period Quarter
 - d. Target Period Year
 - e. Target Period Day
 - f. Year Target

 Note:

Period dimension members in Oracle Hyperion EPM Architect that have the "Data Storage" property set to "Label Only," "Dynamic Calc," or "DynamicCalcandStore " are not displayed.

7. Click **Save**.

Application Mappings

You can define application mappings in cases where you want to define a special period mapping for a specific target application. The mappings that you create here apply to an individual target application.

To create period mappings for an application:

1. On the **Setup** tab, under **Integration Setup**, select **Period Mapping**.
2. Select the **Application Mapping** tab.
3. In **Target Application**, select the target application.
4. Click **Add**.
5. Select the **Period Key**.
6. Enter the following:
 - a. Target Period Month
 - b. Target Period Quarter
 - c. Target Period Year
 - d. Target Period Day
 - e. Year Target
7. Click **Save**.

Source Mappings

Source mappings include explicit and adjustment period mappings. You can create explicit period mappings to ensure that the Oracle Hyperion Financial Data Quality Management, Enterprise Edition periods map correctly to the source system calendar periods. An adjustment period mapping is used only when you select the **Include Adjustment Periods** option when creating the data load rule.

The Source Mapping tab consists of two areas:

- Master—Selects the source system and mapping type.
- Grid—Defines the period mapping. The mapping can be defined only for periods defined on the Global Mapping. New FDMEE periods cannot be created on this tab.

For (Oracle E-Business Suite, Fusion, and PeoplesSoft) source systems, you can select explicit or adjustment systems. For all other systems, you can select only an explicit mapping.



Note:

In Data Rules, you can choose between Default period mapping and Explicit Period mapping. If you choose Period mapping, then source periods are mapped based on the period key and previous period.

To create source mappings:

1. On the **Setup** tab, under **Integration Setup**, select **Period Mapping**.
2. Select the **Source Mapping** tab.
3. In **Source System**, select the source system.
4. In **Mapping Type**, select **Explicit**.

For SAP and JD Edwards source systems, you must select **Explicit** period mappings.

For all other systems (for example, file-based), you can select **Explicit** or **Adjustment**.

5. Click **Add**.
6. Enter the source system **Period Name**, and then click **OK**.



Note:

Period names cannot include spaces if used in a batch script.

7. Enter the source system **Period Key**, and then click **OK**.
8. Enter the source system **Calendar**, and then click **OK**.
9. Enter the source system **GL Period**, and then click **OK**.
The GL Period Number is prefilled based on the Period Name.
10. Enter the source system **GL Name**, and then click **OK**.
11. **Optional:** Enter a description for the mapping.
12. Click **Save**.

To create source period mappings:

1. On the **Setup** tab, under **Integration Setup**, select **Period Mapping**.
2. In **Period Mapping**, select the **Source Mapping** tab.
3. In **Source System**, select the source system.
4. Click **Add**.
5. Click  to select the source system **Period Key**, and then click **OK**.
6. Click  to select the source system **Calendar**, and then click **OK**.
7. Click  to select the source system **Adjustment Period**, and then click **OK**.
8. **For PeopleSoft source systems only:** In **GL Period Year**, enter the general ledger period year.

The General Ledger period year is required for PeopleSoft source systems because PeopleSoft Adjustment Periods definitions do not include a Year value. To properly map adjustment period data from PeopleSoft, define the source accounting period and fiscal year intersections for all PeopleSoft adjustment periods.

9. **Optional:** Enter a description for the mapping.
10. Click **Save**.

To create budget period mappings (for PeopleSoft Commitment Control only):

1. Select **Source Mapping**.
2. In **Source System**, select the source system.

3. Click **Add**.
4. In **Mapping Type**, select **Budget**.

 Note:

From PeopleSoft Commitment Control, only Budget Period data can be extracted. The source calendar/period are based on the control budget definition in PeopleSoft.

5. In **Period Name**, specify the period name.

You can also click  to search for the period name.

 Note:

Period names cannot include spaces if used in a batch script.

6. Enter the source system **Calendar**, and then click **OK**.

You can also click  to search for the calendar name.

7. Enter the source system **GL Period**, and then click **OK**. You can also or click  to search for and select the General Ledger period name.

The GL Period Number is prefilled automatically based on the Period Name.

8. **Optional:** Enter a description for the mapping.
9. Click **Save**.

 Tip:

To delete a mapping, select the mapping, and then click **Delete**.

Defining Category Mappings

You define category mappings for categorizing and mapping source system data to a target EPM Scenario dimension member. For example, in an Oracle Hyperion Financial Management application, you may have a Scenario dimension member called "Actuals" for storing actual balances from a source system. In an Oracle Hyperion Planning application, the same source system data is stored using the Scenario dimension member "Current." In Oracle Hyperion Financial Data Quality Management, Enterprise Edition, you can create one category mapping to give both one name to represent their respective scenarios.

Global Mappings

You can define one global mapping to map various Scenario dimensions to the individual mapping.

The global category mapping lets you define mappings that cross multiple applications. For example, you may have a case where a source category of an actual may map to a target of an actual in the majority of the cases. But you may have a case where you have a target application where the actual may map to current. In this case, it provides the ability to override the global mapping on an application basis.

To define a global category mapping:

1. On the **Setup** tab, and then under **Integration Setup**, select **Category Mapping**.
2. Select **Global Mapping**.
3. Click **Add**.
A blank entry row is displayed.
4. In **Category**, enter the name of the category.
5. In **Description**, enter a description of the category.
6. In **Frequency**, select the frequency of the category.
The category indicates the frequency defined in the period mapping, for example, Daily, Monthly, Quarterly, or Yearly.
7. Enter the target category.
8. Click **Save**.
9. **Optional:** Perform these tasks:
 - To edit a mapping, select the mapping, make changes as necessary, and then click **Save**.
 - To delete a mapping, click **Delete**.

Application Mappings

Unlike global mappings, application mappings can be defined for a target application.

To define application category mappings:

1. On the **Setup** tab, under **Integration Setup**, select **Category Mapping**.
2. In **Category Mappings**, select the **Application Mapping** tab.
3. From **Target Application**, select the target application.
4. Click **Add**.
A blank entry row is displayed.
5. Select the category.
6. Enter the target category or click  to search for a target category.
7. Click **Save**.
8. **Optional:** Perform these tasks:

- To edit a mapping, select the mapping, and then make changes as necessary. Then, click **Save**.
- To delete a mapping, click **Delete**.

3

Integrating Data

Related Topics

- [Loading Data](#)
- [Data Load, Synchronization and Write Back](#)
- [Loading Human Resources Data](#)

Loading Data

Oracle Hyperion Financial Data Quality Management, Enterprise Edition is a solution that allows business analysts to develop standardized financial data management processes and validate data from any source system—all while reducing costs and complexity. FDMEE puts the finance user in total control of the integration process to define source data, create mapping rules to translate data into the required target format, and to execute and manage the periodic data loading process.

Note:

To load data to Oracle Hyperion Financial Management, you must be a valid Financial Management user, and you must be assigned either the "Admin" or "Extended analytics" roles.

Creating Member Mappings

You use member mappings to identify how source dimensionality translates to target dimensionality based on source values. Member mappings are referenced during the data load, enabling Oracle Hyperion Financial Data Quality Management, Enterprise Edition to determine how to dimensionalize the data that is loaded to the target application. They define relationships between source members and target dimension members within a single dimension. You must create a member mapping for each target dimension.

The five types of member mappings:

- **Explicit**—The source value is matched exactly and replaced with the target value.
- **Between**—The range of source values is replaced with a single target value.
- **In**—Enables a list of non-sequential source values to be mapped to one target value.
- **Multi-Dimension**—Target value is assigned for a combination of source segment/chartfields.
- **Like**—The string in the source value is matched and replaced with the target value.

The following table is an example of a member mapping, where three segment members, Cash-101, Cash-102, and Cash-103 map to one EPM member Cash.

Table 3-1 How Segment Members Map to EPM Members

Segment/Chartfield Member	EPM Member
Cash-101	Cash
Cash-102	Cash
Cash-103	Cash
Expense-1	Expense
Expense-2	Expense

You can use special characters for the source values. See [Using Special Characters in the Source Value Expression for Like Mappings](#) and [Using Special Characters in the Target Value Expression](#).



Note:

Target values for multi-dimensional mapping must be an explicit member name. Wildcard or special characters are not supported

To define member mappings:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Mapping**.
2. In **Data Load Mapping**, select the **Location**.
3. Select the **Dimension**.
4. Choose the type of mapping by selecting the Explicit tab, Between tab, Multi-Dimension tab, or Like tab.



Note:

If you define a metadata mapping for the dimension, FDMEE automatically creates a “Like” member mapping. If you enter a member prefix, the same member prefix is automatically entered as the target value in the member mapping. “DEFAULT” displays in the rule name and description field for system-generated mappings. When data is extracted, user-defined mappings are extracted first, and then system generated mappings.

Type options:

- **Explicit**—The source value is matched exactly and replaced with the target value. For example, source value "ABC" is replaced with target value "123." See [Creating Mappings Using the Explicit Method](#).
- **Between**—The range of source values is replaced with a single target value. For example, a range from “001” to “010” is replaced as one value: “999.” See [Creating Mappings Using the Between Method](#).

- **In**—In mappings enable a list of non-sequential source values to be mapped to one target value. In this case, multiple values are mapped to one value within one rule, eliminating the need to create multiple rules (as is required for an Explicit map). For example, you could have source accounts 1503, 1510, and 1515 map to the target account 15000010.
- **Multi-dimension**—For the specified combination of multiple source values a target value is assigned.

For example, if the source value combination is Entity-001,002 Department-ABC, XYZ Account-1222, 1333, then the target value assigned for Account Dimension is 1200.
- **Like**—The string in the source value is matched and replaced with the target value. For example, the source value “Department” is replaced with the target value “Cost CenterA.” See [Creating Mappings Using the Like Method](#).

When processing the source values for transformations, multiple mappings may apply to a specific source value. The order of precedence is Explicit, Between, In, Multi-Dimension, and Like. Within Between and Like types, mappings can overlap.

The rule name determines precedence within a mapping type. Rules are processed in alphabetical order of the rule name within a mapping type. Numbers may also be used to help with ordering. For example, if numbering by tens or one hundreds, you can insert new rules between existing ones. So if rules are numbered 10, 20, and 30, you can add a rule that starts with 25 and need not rename other rules.

Tip:

You can click **Refresh Values** to refresh the list of segment or chartfield values that are displayed. Doing so is helpful when you're creating Explicit mappings.

Creating Mappings Using the Explicit Method

Explicit mappings enable you to enter a source value to be matched exactly and replaced with a target value. Use an explicit mapping to explicitly map the members from the source to a target application. For example, you can map Account1 to the Account100 in your target application. This enables you to explicitly define how to dimensionalize the data file that is loaded into the target application.

To create an Explicit mapping:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Mapping**.
2. From **Dimensions**, select the dimension name.
3. Select the **Explicit** tab.
4. Click **Add**.
5. Enter the **Source Value** or click  to select a value.
See [Using Special Characters in the Source Value Expression for Like Mappings](#).
6. **Optional:** Enter a description for the mapping.

7. Enter the **Target Value** or click  to select a member.
See [Using Special Characters in the Target Value Expression](#).
8. To reverse the sign of the target account specified, select **Change Sign**.
9. In **Description**, specify a description of the mapping.
10. Select **Apply to Rule** to apply the mapping only to the specific data rule in the location.
For other data rules in the location the mappings are not applied.
By default, mappings specified at a location are applicable to all data rules in a location.
11. Click **Save**.

Creating Mappings Using the Between Method

Between mappings enable you to enter a range of source values, separated with a comma. The range of source values is replaced with a single target value. Use a Between mapping to consolidate several accounts from your ledger to a single account in the plan.

To create a Between mapping:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Mapping**.
2. From **Dimensions**, select the dimension name.
3. Select the **Between** tab.
4. Click **Add**.
5. Enter source values in the **Source Value** range.
6. Enter the **Target Value** or click  to select a member.
The target value is the dimension member name. See [Using Special Characters in the Target Value Expression](#).
7. To reverse the sign of the target account specified, select **Change Sign**.
8. Enter the **Rule Name**.
9. In **Description**, enter a description for the mapping.
10. Select **Apply to Rule** to apply the mapping only to the specific data rule in the location.
For other data rules in the location the mapping are not applied.
By default, mappings specified at a location apply to all data rules in a location.
11. Click **Save**.

Creating Mappings Using the In Method

In mappings enable a list of non-sequential source accounts to be mapped to one target account. In this case, multiple accounts are mapped to one account within one rule, eliminating the need to create multiple rules (as is required for an Explicit map).

To create an In mapping:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Mapping**.
2. In **Data Load Mapping**, click **Add**.
A blank row is added.
3. From **Dimensions**, select the dimension name.
4. Enter source values in the **Source Value** range.
The source value is the Enterprise Resource Planning (ERP) segment value. In mappings enable you to specify non-sequential source values. Separate source values with a comma, for example, specify 100,199.
5. Enter the **Target Value** or click  to select a member.
6. To reverse the sign of the target account specified, select **Change Sign**.
7. Enter the **Rule Name**.
8. Enter a description of the In mapping in the **Description**.
9. Select **Apply to Rule** to apply the mapping only to a specific data rule in the location.
For other data rules in the location the mapping are not applied.
By default, mappings specified at a location apply to all data rules in a location.
10. Click **Save**.

Creating Mappings Using the Multi-Dimension Method

Multi-dimension mapping enables you to define member mapping based on multiple source column values. This functionality provides you with the ability to load data into dimensions unavailable in the target application. For example, the mapping for Account dimension can be based on source values of Entity, Product, and Project. Multi-dimension mappings derive the target values based on the combination of source values. In addition, Lookup dimensions can be added to Target Application registration. These dimensions contain source dimensions that do not exist in the target application. They provide even more flexibility in creating multi-dimension filters. It is a way to facilitate conditional data loading.

Note:

When using multi-dimensional mapping, the source needs to be less than or equal to 75 characters.

To create mappings using multiple dimensions:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Mapping**.
2. From **Dimensions**, select the dimension name.
3. Select the **Multi-Dimension** tab.
4. In **Multi Dimension**, click **Edit**.
5. In **Rule Name**, enter the name of the rule.
6. In **Target Value**, specify the target dimension member.

You can also click  to select a target value.

The target values for multi-dimensional mapping must be an explicit member name. Wildcard or special characters are not supported

7. To reverse the sign of the source account value, select **Change Sign**.
8. In **Description**, enter a description of the mapping.
9. Click **Add** to create blank rows for specifying mapping conditions.
10. In **Dimension**, select the dimension to add.

For context, the source Segment/Chartfield column and Data Table columns are shown.

11. In **Condition**, select the method for mapping values.

Available conditions:

- Explicit
- Between
- Like
- In

12. In **Value**, specify the dimension member name.
13. Repeat steps 9-12 to specify multiple conditions.
14. Select **Apply to Rule** to apply the mapping only to a specific data rule in the location.

For other data rules in the location the mapping are not applied.

By default, mappings specified at a location are applicable to all data rules in a location.

15. Click **Save**.

Using Special Characters in Multi-Dimensional Mapping

The Source and Target Value expressions can use special characters. These characters (typically ? and *) can be prefixed or suffixed by one or more characters, which filters the source value by that prefix or suffix.

Special characters include:

- Asterisk (*)—An asterisk (*) represents the source value. The asterisk (*) can be prefixed or suffixed by one or more characters, which filters the source value by that prefix or suffix. The wild card or strips (data load to write back) takes whatever is present in the source and puts it in the target column, usually adding a prefix. An asterisk (*) represents the source value. The asterisk (*) can be prefixed or suffixed by one or more characters, which filters the source value by that prefix or suffix. The wild card takes whatever is present in the source and puts it in the target column, usually adding a prefix.
- Question Mark (?)—The question mark (?) strips a single character from the source value. You can use one or more question marks (?) in the expression. You can also use question marks in combination with other expressions. For example: A?? finds members that start with A and have any two characters following and selects the members or strips off the two characters.

For multiple dimensions, the source dimension is the value brought over and the wild card applies to it alone. The dimensions can be present in a multiple dimensional rule and use wild cards. The prefix/suffix applies only to the source, which equals the target dimension (the dimension on which the rule resides).

Creating Mappings Using the Like Method

Like mappings enable you to enter a string in the source value that is matched and replaced with the target value.

To create a Like mapping:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Mapping**.
2. From **Dimensions**, select the dimension name.
3. Select the **Like** tab.
4. Click **Add**.

A blank row is added.

5. Enter the **Source Value** string.

The source value is the Enterprise Resource Planning (ERP) segment value. Like source write-back mappings support special characters. See [Using Special Characters in the Source Value Expression for Like Mappings](#).

6. Select the **Target Value** or click  to select a member.

The target value is the EPM dimension member name. Like target write-back mappings support special characters. See [Using Special Characters in the Target Value Expression](#).

7. To reverse the sign of the target account specified, select **Change Sign**.
8. Enter the **Rule Name**.
9. In **Description**, enter a description of the Like.
10. Select **Apply to Rule** to apply the mapping only to a specific data rule in a location.

For other data rules in the location the mapping are not applied.

By default, mappings specified at a location apply to all data rules in a location.

11. Click **Save**.

Using Special Characters in the Source Value Expression for Like Mappings

The Source and Target Value expressions can have one or more special characters. Special characters are supported for Like mappings only.

- **Asterisk (*)**

An asterisk (*) represents the source value. The asterisk (*) can be prefixed or suffixed by one or more characters, which filters the source value by that prefix or suffix. The wild card or strips (data load to write back) takes whatever is present in the source and puts it in the target column, usually adding a prefix.

- **Question Mark (?)**

The question mark (?) strips a single character from the source value. You can use one or more question marks (?) in the expression. You can also use question marks in combination with other expressions. For example, A?? finds members that start with A and have any two characters following and selects the members or strips off the two characters.

- **<1>, <2>, <3>, <4>, <5>**

Processes rows that have concatenated values and extracts the corresponding segment value (identified by the segment number). Each segment is separated by an underscore character (_). Only one segment value can be extracted in a source value expression. The source member must use the "_" character as the separator.

 Note:

<1>, <2>, <3>, <4>, <5> can be used with a question mark (?) but cannot be used with an asterisk (*).

- **<BLANK>**

Processes only rows that contain the blank character (space).

The system only reads the expression where the source member is ' ' as <BLANK>. In this case, single quotes surround a single space character. If the source has NULL, which is shown like ,, or as a space surrounded by " ", then the system does not interpret the NULL as a <BLANK>. Only the '<space char>' expression is interpreted.

 Note:

The <BLANK> notation may be used in both source and target expressions. If used in a target expression, it writes a blank space to the target.

Use the <BLANK> target when you need to write back to the Peoplesoft journal interface table for any blank fields.

This is true for both single and concatenated segment or chartfield dimension mappings.

Table 3-2 Examples of Expressions Using Special Characters

Special Character(s) Used	Mapping Type	Source Value	Target Value	Result	Notes
*	Data Load	*	1000	1000 returns 1000 WXYZ returns 1000	In this example, Oracle Hyperion Financial Data Quality Management, Enterprise Edition processes all rows and overrides the source value with a default value of 1000. In this expression, WXYZ also returns 1000. Because you entered an asterisk for the source value FDMEE replaces any source value with the target value of 1000.
*	Data Load	*	*	1000 returns 1000 WXYZ returns WXYZ	In this example, FDMEE processes all rows and replaces the source value as is.
*	Stripping	*	A*	101 returns A101	Processes all source members, and adds an "A" as a prefix.
*	Stripping	*_DUP	*	1000_DUP returns 1000	Processes and strips off only source values ending with "_DUP."
?	Stripping	?*	*	A1000 returns 1000 B2000 returns 2000	This result processes only source values of one or more characters in length. Strips off the first character

Table 3-2 (Cont.) Examples of Expressions Using Special Characters

Special Character(s) Used	Mapping Type	Source Value	Target Value	Result	Notes
?	Stripping	*????	*	1000_DUP returns 1000 A1000 returns A	This result processes only source values of four or more characters in length. Strips off the last 4 characters
<1>, <2>, <3>, <4>, <5>	Data Load	<1>	*	01_420 returns 01	
<1>, <2>, <3>, <4>, <5>	Data Load	<2>	*	01_420 returns 420	
<1>, <2>, <3>, <4>, <5>	Data Load	<3>	*	01_420_AB_CC1_001 returns AB	
<1>, <2>, <3>, <4>, <5>	Stripping	?<1>	*	A01_420 returns 01	
<BLANK>	Data Load	<BLANK>	[None]	' ' returns [None] '01_ ' returns [None]	Single quotation marks are shown for illustration only.

Automap Wildcarding

Oracle Hyperion Financial Data Quality Management, Enterprise Edition enables target-account or target-entity derivation by permitting wildcard characters (*) and (?) in source and target members. Mapping-table records that have wildcard characters in the source and target column are considered automapped.

FDMEET does not validate the target value.

Example Automap

Table 3-3 Automap Example

Rule Name	Rule Description	Source Value	Target Account
w0011--	Cash Accts	0011??	Cash.??

Example General Ledger Trial Balance Records

Table 3-4 Example of General Ledger Trial Balance

GL Account	Center	Description	Amount
001100	0160000	Cash In Bank	1000.00
001101	0000000	Cash Corp LB	2000.00

Table 3-4 (Cont.) Example of General Ledger Trial Balance

GL Account	Center	Description	Amount
001116	0001000	Petty Cash	1000.00
223500	0160000	AP	5000.00

Resulting Record Conversion

Table 3-5 Result of Record Conversion

GL Account	Hyperion Account
001100 0160000	Cash.00
001101 0000000	Cash.01
001116 0160000	Cash.16

Explanation of the Conversion Process

The criteria for the Automap entry (Like 0011??) retrieves the first three records from the general ledger trial balance. Because the Automap entry contains wildcard characters in the target account column, Oracle Hyperion Financial Data Quality Management, Enterprise Edition must search the source account to replace the wildcard characters within the target account with actual characters from the source account.

The source-account characters represented by ?? marks (under Source Value) are used to replace the two question marks that follow cash (under Target Account).

Conditional Mapping using a Mapping Script

With conditional mapping, source members are mapped to script expressions rather than to hard-coded target members. Conditional mapping is valid only for rule-based mapping (Between, In, and Like). You can activate conditional mapping by placing #SCRIPT or #SQL in the Target value column. Use #SCRIPT for Jython Script and #SQL for SQL script. Conditional mapping, in conjunction with dimension processing order, enables mapping that is based on the results of dimension mappings. That is, dimension mappings that have already been processed. See [Using Mapping Scripts](#).

Note:

In Oracle Hyperion Financial Data Quality Management, Enterprise Edition, Jython script is not supported for conditional mapping (#SCRIPT cannot be used in the Target value column.)

To apply conditional mapping:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Mapping**.
2. In **Data Load Mapping**, click **Add**.
A blank row is added.

3. From **Dimensions**, select the dimension name.
4. Select either the **Between**, **In**, or **Like** tab.
5. Enter source values in the **Source Value** range.
6. In **Target Value**, enter **#SQL**.
To use Jython script, enter **#SCRIPT**. For more information, see [Using Jython Objects with Mapping Scripts](#).

The pencil icon is enabled.

Data Load Mapping [Save] [Cancel]

Dimensions: Account Data Table Column: ACCOUNT

View: [Add] [Delete] [Detach] [Validate] [Refresh Values]

Source Value	Target Value	Script	Change Sign	Rule Name	Description
*	#SQL	[Pencil Icon]	[X]	wx	

7. Click the pencil icon.
8. On the **Edit Script** screen, specify your SQL script.
This sample script uses the SQL Cast statement to conditionally process assigned values for the target column.

Edit Script [X]

Script

```
#
# Sample SQL script. Script uses SQL CASE statement to conditionally
# process assign value for the target column.
#
CASE
  WHEN ACCOUNT LIKE '61%' AND ud1 = '205' THEN '6110'
  WHEN ACCOUNT LIKE '61%' AND ud1 = '240' THEN '6120'
  ELSE '6130'
END
```

[OK] [Cancel]

FDME does not perform an error check or validate the script. You need to test the script on your data files in a test environment and verify the results.

9. In **Rule Name**, specify the data load rule to use with the mapping script.
10. Click **Save**.

Using Special Characters in the Target Value Expression

You can use only an asterisk (*) in the target expression, and you can prefix or suffix any number of characters to the asterisk (*) character. When you run the rule, the asterisk (*) is replaced by the resulting source value (which may or may not have its own source expression), and is concatenated to any prefix or suffix that you have specified in the target expression. For example:

Target Value:

A*

Result:

1000 = A1000

Target Value:

*_DUP

Result:

1000 = 1000_DUP

 Note:

<BLANK> is supported in the target value expression in data mappings and can be used in all mapping types (Like, Between, Explicit, and Multi-dimension). When writing data to an Enterprise Resource Planning (ERP) GL interface table, the <BLANK> notation may be used for a target dimension mapping in order to successfully pass the validation step in the workflow process. For example, when writing back to the Peoplesoft journal interface table, the specification of <BLANK> can be used when the user does not want to provide a value for a chart field value, but needs to successfully validate the write-back data.

Format Mask Mapping for Target Values

Oracle Hyperion Financial Data Quality Management, Enterprise Edition supports the ability to specify a format mask for a target member. The format mask defines the target member based on a combination of the source member details, and optional user defined text. This functionality is useful when designating the target member based on some part of the source member, an additional prefix, suffix, or replacement text for the target.

The format mask is available for the target member specification for all mapping types except explicit. Common usage of this mapping type falls into three categories: replacing segments from the source, replacing segments with string operations, and replacing segments with string operations using a prefix or a suffix.

#FORMAT Mapping Type Components

The #FORMAT mapping type consists of the following components:

Table 3-6 #Format Mapping Type Components

Component	Description
#FORMAT	Indicates that a mapping type of FORMAT is specified in the target member.

Table 3-6 (Cont.) #Format Mapping Type Components

Component	Description
<format mask>	User defined format mask with the following characters used to define the format: • "?"—Include a character from a specific position in the source member or segment within a member. • "#"—Skip or drop a character from the source when creating the target member. • "character"—Include the user defined character on the target "as- is". Used for prefixing, suffixing or any fixed string or required character. This can be used in conjunction with the special format mask characters. • "*"—Include all characters from the source segment or source. When "*" is used as the only format mask character in a segment, then the entire segment value is copied from the source. When "*" is used in conjunction with "#" or the "?" character, then all remaining and unused characters are brought over. "*" is a wildcard character that takes the remaining characters not specified by "?" or "#". For example, when the source is "abcd" and "*" is used, then the target is "abcd." When the target is "?#*", then the result is "acd." If Oracle Hyperion Financial Data Quality Management, Enterprise Edition encounters a "*" within a segment, then anything specified after the "*" is ignored other than the "character" specified on the format.
<segment delimiter>	The optional segment delimiter defines the character that is used to delimit the segments in the source and target member. For this rule type, the source and target delimiter must be the same. When the segment delimiter is not specified, then the format mask is applied to the entire member independent of any segment specification or delimiter.

#FORMAT Mapping Example

The following is an example that uses all options provided by #FORMAT:

Table 3-7 #Format Mapping Type Example

Source	Target	Result
12345-6789-012-3456ABC-001	#FORMAT("???*-GROUP-AA##? #*X-GROUP","-") Explanation: Take the first three characters of the first segment, take the entire second segment, replace the third segment with the text "GROUP," prefix the fourth segment with AA, drop the third and fourth characters, keep the fifth character, drop the sixth character, keep ABC and add suffix "X, replace the fifth segment with the text "GROUP."	123-6789-GROUP-AA5ABCX- GROUP

Replacing Segments

You may want to use the format of the source member as the definition of the target member, but replace some of the source segments rather than reuse the values from the source. For example, you may have a requirement to filter the source by the value of the 4th segment, replace the 7th segment with an explicit value, and then retain the values of the other segments as in the following:

Source:

```
?????-?????-?-012000000-?????-???-??????-?????-?????-?????-???
```

Target:

```
?????-?????-?-012000000-?????-???-GROUP-?????-?????-?????-???
```

Replacing Segments with String Operations

You may also need to perform a string operation on a segment that is being replaced. For example, you may have a value of 11002293, but when the segments are written, you want to only take the last four digits, or the first six digits. Examples using the member 11002293:

- Ignore the first two characters and provide the result: 002293. Use #FORMAT("##*").
- Truncate the last three characters provide the result: 11002. Use #FORMAT("?????").
- Ignore the first two and truncate the last three with the result: 002. Use #FORMAT("##???").

Replace Segments with String Operations and Using an Prefix or Suffix

You may want to use the segment value from the source as-is in the corresponding segment in the target. For example, if the source is A100, you can map the value as the value in the target, and you can map this value as the value in the target. In this case, use a wildcard on the source, and then specify the explicit value for that segment in the target based on the source.

**Note:**

If any other string operation is desired, then use scripting.

Using the #FORMAT Mapping Type

To use the #FORMAT mapping type:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Mapping**.
2. In **Dimensions**, select the dimension name.
3. Select the **Between**, **In**, or **Like** tab.
4. Click **Add**.
5. In **Source Value**, enter the segment to map.
For example, on the **Like** tab, enter: 12345-6789-012-3456ABC-001 in the Source Value field.
6. Select the format map for the target member using the #FORMAT(<format mask>, <segment delimiter>).
For example, enter #FORMAT("???-* -GROUP-AA##?*X-GROUP", "-").
See [#FORMAT Mapping Type Components](#).
7. To reverse the sign of the target account specified, select **Change Sign**.
8. Enter the **Rule Name**.
9. In **Description**, enter a description of the mapping.
10. Select **Apply to Rule** to apply the mapping only to a specific data rule in a location.
By default, mappings specified at a location apply to all data rules in a location.
11. Click **Save**.

The result of applying format map created in steps 5 and 6: 123-6789-GROUP-AA5ABCX-GROUP.

Ignoring Member Mappings

You can ignore loading data to a particular dimension member.

To ignore member mappings:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Mapping**.
2. Select a source value, and in **Target Value**, enter **ignore**.
For example, assume that a business user does not require extraction of data relating to Departments 101, 103 and 105. You specify an **In** mapping with the source values, 101, 103, and 105, and then for the target value, you specify **ignore**. In this way, data relating to Departments 101, 103 and 105 is extracted, but not written to the application in the Import Format option.

Importing Member Mappings

You can import member mappings from a selected .CSV and .TXT file, and then you can create mappings. Import member mappings support merge or replace modes, along with validate or no validate options for target members.

Importing member mappings can be executed in either online or offline mode.

You can also import mappings from Excel or download an Excel template.

. See [Importing Excel Mappings](#) and [Downloading an Excel Template \(Mapping Template\)](#)

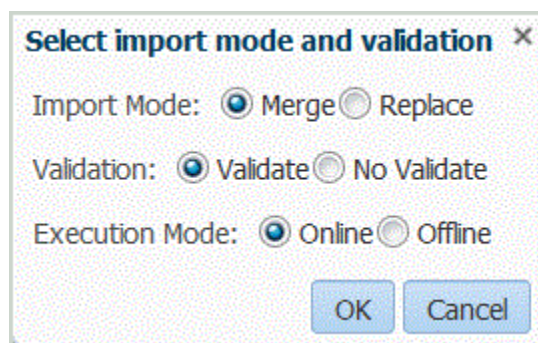
To import member mappings:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Mapping**.
2. From the **Import** drop-down, select one of the following:
 - Current Dimension
 - All Dimensions
 - Import from Excel
See [Using Excel Trial Balance Files to Import Data](#).
 - Download Excel Template
See [Downloading an Excel Trial Balance Template](#).

The Select file to import screen is displayed.

3. Navigate to the file to import and click **OK**.
4. **Optional:** If necessary, click **Upload** to navigate to the file to import, and then click **OK**.

The Select Import Mode and Validation screen is displayed.



5. From **Import Mode**, select the import mode:
 - Merge—Overwrites the data in the application with the data in the data load file.
 - Replace—Clears values from dimensions in the data load file and replaces it with values in the existing file.
6. From **Validate**, select to validate the member mappings.

Validate ensures that all data in the imported General Ledger has a corresponding mapping.

7. In **Execution Mode**, select the mode for executing the import:

- Online—Processes the import immediately.
- Offline—Runs the import in the background.

8. Click **OK**.

In the member mapping import files, Oracle Hyperion Financial Data Quality Management, Enterprise Edition supports one of the following characters as column separators:

- ,
- |
- ;

The order of the columns:

- source value
- target value
- rule name
- rule description



Note:

If you add a minus sign in front of a target account value, then it is imported with the “Change Sign” selected.

Table 3-8 Mapping Source Types

Column	Mapping
100, Cash, 100, Explicit Mapping	Explicit Mapping
100>199, Cash, R2, Between Mapping	">" indicates its BETWEEN mapping.
1*, Cash, R3, Like Mapping	"*" indicates its LIKE mapping.
#MULTIDIM ACCOUNT=[4*] AND UD3=[000],Cash,R4,Multi Dimension Mapping	"#MULTIDIM" indicates a multiple dimension mapping. The actual column name used for the mapping is the Data Table Column Name. The easiest way to create a multiple dimension mapping is to create a mapping through the user interface and export it to the file. You can then modify the file by applying additional mapping.
10, 20, In Mapping	Source values are enclosed with " " and separated by a comma (,) for the In mapping. For example IN 10, 20 is defined as "10,20" in the source column of the import file.

Downloading an Excel Template (Mapping Template)

In Data Load Mapping using the import feature, you can select and import an Excel mapping, and specify whether to merge or replace the mappings. Excel mapping templates with correct formatting are included in the `EPM_ORACLE_HOME/products/FinancialDataQuality/templates` directory.

The mapping template also includes a macro script that pulls Oracle Hyperion Financial Management dimensions directly from the target application to which you are connecting.

You must upload the Excel template to the Oracle Hyperion Financial Data Quality Management, Enterprise Edition server, and then pick the excel file as the file to load in the data load rule, or when prompted by the system if the file name is left blank. The system determines if the file being processed is an Excel file, and then reads the required formatting to load the file.

When working with a mapping template in Excel:

- Do not have any blank lines in the map template.
- You can insert lines in the template, but you have to keep new lines in the designated area.
- Each template supports a single dimension.

To download an Excel template:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Mapping**.
2. Select the **All Mapping** tab.
3. In the **Import** drop-down, select **Download Excel Template**.

A Maploader.xls file is downloaded. Copy or save the file to your hard drive.

4. Open the Maploader.xls file.
5. Select the **Map** tab.
6. Enter the **Location** in **cell B1**, **Location ID** in **cell B2**, and select the dimension from the **Dimension** drop-down in **cell B3**.
7. Complete the following column fields:
 - a. In **Source**, enter the source dimension value.

You can specify wildcards and ranges when entering the source dimension.

- Wildcards for unlimited characters—Use asterisks (*) to denote unlimited characters. For example, enter 548* or *87.8.
- Wildcards for single character place holders—Use questions marks (?) to denote single character place holders. For example,
 - 548??98
 - ??82???
 - ??81*
- **Range**—Use commas (,) to denote ranges (no wildcard characters are allowed). For example, specify a range as 10000,19999.

(This range evaluates all values from 10000 to 19999 inclusive of both start and end values.)

In this case, FDMEE considers all values from 10000 to 19999 to include for both start and end values.

- **In map**—Use commas (,) to separate entries (no wildcard are characters allowed). You must have at least three entries or the map shows as a between map. For example, specify an In map as 10,20,30.
 - **Multi-Dimension map**—Use #MULTIDIM to indicate its multidimensional mapping. Enter the DIMENSION NAME=[VALUE] and the value. The value follows the logic as wildcard, range, and In map. In the following example the search criteria is all accounts starting with 77 and UD1 = 240. For example, #MULTIDIM ACCOUNT=[77*] AND UD1=[240].
- b. In **Source Description**, enter a description of the source value.
 - c. In **Target**, enter the target dimension value.
 - d. In **Change Sign**, enter **True** to change the sign of the Account dimension. Enter **False** to keep the sign of the Account dimension. This setting is only used when mapping the Account dimension.
 - e. In **Data Rule Name**, enter the data rule name when the mapping applies to a specific data rule name.

Note:

If you are adding an Explicit mapping, the rule name must equal the source value.

	A	B	C	D	E
1	FDM Location:	TEXAS			
2	FDM Location ID:	751			
3	Dimension:	Account			
4					
5					
6	Source	Source Description	Target	Rule Name	Convert (True/False)
7		IDataMap	TargKey		ChangeSign
8		SrcDesc			
9	1*	Acct Like 1		*w1x9	FALSE
10	4110,4120,4140	Acct in 4110,4120,4140	4110/w411010		FALSE
11	6*	Acct in 6 range	6110/w6x11		FALSE
12	7000,7999	Acct in 7000 range	#SCRIPT/w700012		FALSE

Importing Excel Mappings

You can import Excel mappings by selecting the Import option and selecting an Excel mapping.

Note:

The import of mapping rules using an Excel template provides a place to specify a mapping script.

To import an Excel mapping:

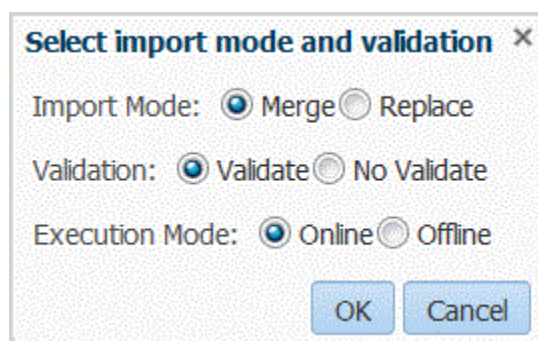
1. On the **Workflow** tab, under **Data Load**, select **Data Load Mapping**.
2. Select the **All Mapping** tab.
3. In the **Import** drop-down, select **Import from Excel**.
4. In **Select a file to import**, select the Excel file to import, and then click **OK**.

**Note:**

If you are importing an Excel 2010 or 2016 file that has already been exported, open the file before importing it. This step launches macros in the Excel file that are required for the import process.

5. **Optional:** If necessary, click **Upload** to navigate to the file to import, and then click **OK**.

The Select Import Mode and Validation screen is displayed.



6. From **Import Mode**, select the import mode:
 - Merge—Overwrites the data in the application with the data in the Excel data load file.
 - Replace—Clears values from dimensions in the Excel data load file and replaces it with values in the existing file.
7. From **Validate**, select to validate the member mappings.

Ensures that all data in the imported General Ledger has a corresponding mapping.
8. In **Execution Mode**, select the mode for executing the import:
 - Online—Process the import immediately.
 - Offline—Runs the import in the background.
9. Click **OK**.
10. Click **OK**.

The mapping inherits the default data load rule, and shows the description of "System Generated Mappings."

If you use Explicit mapping, the data rule name must equal the source value.

Exporting Member Mappings

You can export member mappings to a selected file, or an Excel file.

To export member mappings:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Mapping**.
2. Select **Export**.
3. In the **Export** drop-down, select a method:

Export options:

- Current Dimension
 - All Dimensions
 - Export to Excel
4. From the **Specify file location** for **Current Dimension** and **All Dimensions** export methods, specify the file name in **File Name**, or navigate to the file to export, and then click **OK**.

For the **Export to Excel** method, mappings are exported to a Microsoft Excel spreadsheet. Open or save the XLS file as desired.

When you export to Excel, you cannot re-import in that format.

When the file has been exported, Oracle Hyperion Financial Data Quality Management, Enterprise Edition displays the message: "File exported successfully."

5. **Optional:** Click **Upload** or **Download** and navigate to the file to export, and then click **OK**.

Deleting Member Mappings

You can delete all member mappings or only those mappings for which there is a tab in which mappings have been added. You can delete all the mappings in the dimension you are currently on, or just the row.

To delete member mapping from a selected tab:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Mapping**.
2. Select the tab from which to delete mappings.

For example, select the **Explicit** tab to view explicit type mappings.

To view all mappings, select the **All Mappings** tab.

3. Select the mapping and click **Delete Mappings**.

To delete multiple mappings, use the **Shift** key to select multiple mappings.

To delete all mappings, use **Ctrl+A** key.

4. In **Are you sure you want to delete the selected data load mapping(s)**, click **OK**.
5. Click **Save**.

 Note:

To delete all mappings, select "Delete All Mappings."

Restoring Member Mappings

Restoring member mappings deletes mappings made in the current session and restores mappings based on the point of view.

You can restore member mappings only from the last data load for the POV.

To restore a member mapping:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Mapping**.
2. Select **Restore Mapping**.
3. In **Restore Mapping Confirmation**, click **OK**.

Defining Data Load Rules to Extract Data

After you define member mappings for the data load rule, define data load rules for ledgers or business units in your source system. Data load rules allow you to create an integration definition that can be reused for each period. They use a point of view specified by a user for a period and category. Data load rules are defined for locations that you have set up. Data load rules are specific to:

- locations
- Ledgers for Fusion and E-Business Suite source systems
- Business units for PeopleSoft Enterprise Financial Management source systems

You can create multiple data load rules for a target application so that you can import data from multiple sources into a target application. Use the following high-level process to create a data load rule:

1. Create the data load rule.
2. Define data load rule details.
3. Execute the data load rule.

Defining Data Load Rule Details

You create and modify data load rules on the Data Load screen. The Data Load Rule screen window sections:

- Data Rule Summary
- Data Load Details
- Source Filters, which consist of four tabs: Source Options, Target Members (for file-based data load rules in which a driver is specified), Target Options, and Custom Options)

See [Working with Target Options](#) (by location) and [Registering Target Applications](#).

See [Creating Custom Options](#).



Note:

Before you create data load rules, ensure that your source system data does not include special characters, which are not supported in Oracle Hyperion Financial Management target applications.

To define the data load details for a source system:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. In the **Data Load** summary area, click **Add**.
3. In **Details**, in **Name**, enter the data load rule name.
4. Select a **Category**.

The categories listed are those that you created in the Oracle Hyperion Financial Data Quality Management, Enterprise Edition setup. See [Defining Category Mappings](#).

5. In **Period Mapping Type**, select the period mapping type for each data rule.

Valid options:

- **Default**—The Data Rule uses the Period Key and Prior Period Key defined in FDMEE to determine the Source General Ledger Periods mapped to each FDMEE period included in a Data Rule execution.
- **Explicit**—The Data Rule uses the Explicit period mappings defined in FDMEE to determine the source General Ledger Periods mapped to each FDMEE Period included in a Data Rule execution. Explicit period mappings enable support of additional General Ledger data sources where periods are not defined by start and end dates.
- **None**—With source adaptors use this option to ignore source period mappings. Thus, all imported data rows are mapped to the FDMEE period selected in a Data Rule execution.

These options are unavailable for a file-based definition.

Table 3-9 Period Mapping Types

Location Type	Import Format Type	Period Mapping Default	Explicit Period Mapping Explicit	Period Mapping	Include Adjustment Periods
With Accounting Entity	Standard	Yes	Yes	N/A	Yes
Without Accounting Entity	Standard	Yes	Yes	N/A	Yes
With Accounting Entity	Source Adapter	N/A	N/A	Yes	Explicit: Yes None: N/A
Without Accounting Entity	Source Adapter	N/A	N/A	Yes	Explicit: Yes None: N/A

6. **Optional:** Enter a description.

7. From **Target Plan Type**, select the plan type of the target system.
8. Select the source options.

The following options may appear in the Details section or the Source Options section depending on the source system.

- **Target Plan Type** (Oracle Hyperion Planning and Oracle Essbase)—Select the plan type which contains only the dimensions, members, and data values relevant to that plan type. The plan type applies to the source or target system depending on the POV location for this data load rule.

FDMEE supports data loads for up to six plan types (including custom and Planning applications.)

- **Zero Balances**—For SAP and JDE, select the zero balances option:
 - **Include**—Includes a zero balance for any reason.
For example when there is a debit of 5 and a credit of 5, then the zero amount is included.
 - **Exclude No Activity**—Zero balances are excluded when the beginning balance debit, beginning balance credit, period debit and period credit equal 0 (begin_bal_dr, begin_bal_cr, period_dr, period_cr all have 0 for the YTD balance type, or the period debit and the period credit have a period for the period balance type (period_dr, period_cr equals 0 for the Periodic balance type).
 - **Exclude Zero Net Balance**—Zero net balances are excluded when the beginning balance debit minus the beginning balance credit plus the period debit minus the beginning credit plus period debit minus the period credit equals 0 for the Year to Date balance type (begin_bal_dr – begin_bal_cr + period_dr – period_cr = 0 for the YTD balance type, or the period debit minus the period credit equals zero (period_dr – period_cr = 0 for the Periodic balance type).

The following example shows how each include zero balance option affects account balances.

Table 3-10 Zero Balance options

Amount	Opening Balance	Transaction Debit	Transaction Credit	Closing Balance
4000	40000	0	40000	0
5000	50000	25000	75000	0
6000	0	0	0	0
7000	0	35000	35000	0

When the Zero Balance is "Include," Accounts 4000, 5000, 6000 and 7000 qualify for this condition because all zero balanced accounts are included.

 Note:

The Include Zero Balance option is not applicable when pulling data from Peoplesoft.

When the Zero Balance is "Exclude No Activity" is selected, only Account 6000 is excluded because the Opening, Transaction and Closing balances are all zero and there is no activity. Accounts 4000, 5000, and 7000 are extracted.

When the Zero Balance is "Exclude Net Zero" is selected, Accounts 4000, 5000, 6000 and 7000 are excluded because their closing balance is zero.

- Include Adjustment Periods—Select to include adjustment periods.

Adjustment periods ensure that the FDMEE adjustment periods map correctly to the source system adjustment periods.

When you explicitly map period 13 to December/Period 12, and select the Include Adjustment Period option, then the following occurs:

- For YTD balances, period 13 becomes the ending balance.
- For PTD balances, period 13 and December/Period12, are added.

To define source options:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. In **Data Load Rule**, select a data load rule or click **Add**.
3. Select the **Source Options** tab.
4. Complete the source options or source filter options based on the source system:

Source filter option:

- For Fusion and E-Business Suite source systems, see [Defining Source Filter Options for E-Business Suite Source Systems](#).
 - For PeopleSoft Enterprise Financial Management source systems, see [Defining Source Filter Options for PeopleSoft Enterprise Financial Management Source Systems](#).
 - For PeopleSoft human resource source systems, see [Defining Human Resource Data Load Rules](#).
 - For JD Edwards GL, see [Defining Source Filter Options for JD Edwards GL Source Systems](#).
 - For SAP, see [Defining Source Filter Options for SAP Adapters](#).
 - For the Open Interface Adapter, see [Defining Source Filter Options for the Open Interface Adapter](#).
 - For Financial Management source systems, see [Defining Source Filter Options for Financial Management](#). For information about Financial Management source parameters, see [Defining Source Parameters for Financial Management](#).
 - For file-based source systems, see [Defining Data Load Rule Details for a File-Based Source System](#).
5. **Optional:** If you are working with a multi-column data load, select the **Column Headers** tab, and specify the start date and end date of the numeric columns.
See [Loading Multi-Column Numeric Data](#).
 6. **Optional:** To work with target options, select the **Target Options** tab, and select any options.
 7. **Optional:** You can specify free form text or a value by selecting **Custom Options** and specifying the text you want to associate with the data load rule.

See [Creating Custom Options](#).

8. Click **Save**.

Defining Source Filter Options for E-Business Suite Source Systems

When defining data load mapping details, you can define the data to extract, including whether or not to extract:

- The amount type—Only monetary, statistical, or both, monetary and statistical amounts
- Zero balance accounts where the debits and credits for an account total zero and there is not period activity.
- Adjustment periods—Determines whether to extract balances in the adjustment period
- Standard or Average balances—Average balances contain only balance sheet data.
- Source balance type—Actual, Budget, or Encumbrance

In Oracle Hyperion Financial Data Quality Management, Enterprise Edition, you classify the data to transfer with the valid types in the source accounting entity of Actual, Budget, and Encumbrance. Typically, you do not map a segment from the chart of accounts to the Scenario dimension, so you choose a default member as part of the data rule definition.

You can extract functional balances, which are stored in the base currency of the selected ledger or business unit. For example, when transactions are entered in multiple currencies, the total of all transaction balances is expressed in the functional currency.

You can also extract entered balances, which are balances associated with a currency attached to a primary ledger other than functional/local currency.

Additionally, FDMEE can import the data in a currency specified by the user. In this case, the balances must be translated to the specified currency in the source system. This can be achieved by running the Currency Translation process in the source Enterprise Resource Planning (ERP) system. (FDMEE does not perform any currency translations.)

Oracle E-Business Suite currently does not provide a page to call for displaying commitment data.

To define the data load source filter for Fusion and E-Business Suite source systems:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. In **Data Load Rule**, select a data load rule or click **Add**.
3. Select the **Source Options** tab.
4. In **Accounting Entity**, specify the accounting entity from the list of values of the source system.

You can select the accounting entity in this field, or when entering location detail. Data rules in locations without an accounting entity require you to select an Accounting Entity.

This functionality applies only to data rules in a location using a standard import format.

You cannot modify the accounting entity once the Data Rule has been executed.

5. In **Accounting Entity Group**, specify the accounting entity group name if the location is associated with an accounting entity group.

When a data rule in a location includes an accounting entity, then the rule is constrained by the accounting entity in the definition. In this case, the data rule in the location cannot use an accounting entity group.

6. In **Include Adjustment Periods**, select **yes** or **no**.

Adjustment periods ensure that FDMEE adjustment periods map correctly to the source system adjustment periods.

When you explicitly map period 13 to December/Period 12, and select the Include Adjustment Period option, then the following occurs:

- For YTD balances, period 13 becomes the ending balance.
- For PTD balances, period 13 and December/Period12, are added.

7. Select the **Amount Type**:

- Monetary
- Statistical—The balance selection of entered or functional currency does not apply.
- Monetary and Statistical

8. From **Currency Type**, select the currency type by which to extract balances:

9. In the **Zero Balances** drop-down, select the zero balances option:

- Include—Includes a zero balance for any reason.
For example, when there is a debit of 5 and a credit of 5, then the zero amount is included.
- Exclude No Activity—Zero balances are excluded when the beginning balance debit, beginning balance credit, period debit and period credit equal 0 (begin_bal_dr, begin_bal_cr, period_dr, period_cr all have 0 for the YTD balance type, or the period debit and the period credit have a period for the period balance type (period_dr, period_cr equals 0 for the Periodic balance type).
- Exclude Zero Net Balance—Zero net balances are excluded when the beginning balance debit minus the beginning balance credit plus the period debit minus the beginning credit plus period debit minus the period credit equals 0 for the Year to Date balance type (begin_bal_dr – begin_bal_cr + period_dr – period_cr = 0 for the YTD balance type, or the period debit minus the period credit equals zero (period_dr – period_cr = 0 for the Periodic balance type).

The following example shows how each include zero balance option affects account balances.

Table 3-11 Zero Balance options

Amount	Opening Balance	Transaction Debit	Transaction Credit	Closing Balance
4000	40000	0	40000	0
5000	50000	25000	75000	0
6000	0	0	0	0

Table 3-11 (Cont.) Zero Balance options

Amount	Opening Balance	Transaction Debit	Transaction Credit	Closing Balance
7000	0	35000	35000	0

When the Zero Balance is "Include," Accounts 4000, 5000, 6000, and 7000 qualify for this condition because all zero balanced accounts are included.

When the Zero Balance is "Exclude No Activity" is selected, only Account 6000 is excluded because the Opening, Transaction, and Closing balances are zero and there is no activity. Accounts 4000, 5000, and 7000 are extracted.

When the Zero Balance is "Exclude Net Zero" is selected, Accounts 4000, 5000, 6000, and 7000 are excluded because their closing balance is zero.

- 10. In Signage Method**, select the method to flip the sign of amounts when data is loaded.

Available methods:

- **Absolute**—Loads the data based on the following rules:

Table 3-12 Absolute Signage rules

Account Type	GL (GAAP)	EPM (Absolute)
Revenue	naturally negative	signage flipped
Liability	naturally negative	signage flipped
Equity	naturally negative	signage flipped
Expense	naturally positive	signage unchanged
Asset	naturally positive	signage unchanged

- **Same as source**—Loads the same sign as recorded in the source system.
 - **Reverse from source**—Loads the reverse of the sign as recorded in the source system.
- 11. Select the Amount for Balance Sheet Accounts and Amount for Income Statement Accounts:**
- **YTD**—Year-to-date account balance, where account balances accumulate from the beginning of the year to the current period. Typically, balance sheet accounts (assets, liabilities, and equities) are specified with a YTD balance. (The default is YTD.)
 - **Periodic**—Account balance for the specific period. Typically, income statement accounts (revenues and expenses) are specified with a periodic balance. (The default is PTD.)
- 12. From Currency Type**, select the currency type by which to extract balances:
- **Functional**—Balances stored in the base currency of the selected ledger or business unit (local currency)
 - **Entered**—Balances associated with a currency attached to primary ledger other than the functional/local currency that you specify in the Currency Code field.
 - **Translated**—FDMEE can import the data in a currency specified by the user. In this case, the balances must be translated to the specified currency in the source system. This task can be achieved by running the Currency Translation

process in the Enterprise Resource Planning (ERP) system. (FDMEE does not perform any currency translations.) Additionally, you need to specify the translated currency code in the Currency Code field.

13. From **Currency Code** (Entered and Translated currency types only), select the ISO 4217 currency code to use with an entered or translated currency type.

For example, enter EUR to select the EURO currency code.

14. Select the **Balance Method**:

- **Standard**—In Oracle General Ledger, accounting transaction balances are stored as-is, also known as standard balances.
- **Average**—Average balances only contain balance sheet data. If you selected Statistical as the amount type, the Balance Method is ignored.

15. Select the balance type to extract:

- **Actual**
- **Budget**—If you select the Budget source balance type, click **Add** to select budget types to include in the extraction.
- **Encumbrance**—If you select the Encumbrance source balance type, click **Add** to select encumbrance types to include in the extraction.

16. Select the segment values to extract.

If you do not want to extract all data from the source general ledger, filter data by the balancing segments of the source. Options:

- **All**
- **Selected**

In Oracle E-Business Suite, the balancing segment ensures that at this level, balancing debits equal credits. When you create a data load rule, you can extract the general ledger balances relating to all the members of the balancing segment or for specific members of it.

To select the balancing segment values, click , and then select segment values, and then click **OK**.

To deselect a value, click , and then in the **Select Balancing Segment Values** dialog box, clear any values, and then click **OK**.

17. Select the **Beginning**, **Ending**, and **Average** exchange rate options.

Enterprise Resource Planning (ERP) source systems maintain comprehensive exchange rate information for transaction processing. Target applications can use this information by extracting the exchange rates. You can select a beginning, ending, and average rate type from the source system. (The types in the source system may not explicitly define those rates types but are mapped to the rates types in the FDMEE interface table.)

 Note:

For Oracle Hyperion Planning applications, exchange rates are loaded only when the "Classic" data load method is selected.

**Note:**

You define exchange rate options only when the target application is multi-currency.

**Note:**

For Account Reconciliation Manager applications, you must not choose multiple rate options.

18. Click **Save**.

19. Define the target filter options.

After you define the target filter options, run the data rule. See [Running Data Load Rules](#).

Defining Source Filter Options for PeopleSoft Enterprise Financial Management Source Systems

When defining data load rule details, you can specify various data extraction options as described below.

To define the source filter options for PeopleSoft Enterprise Financial Management source systems:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. In **Data Load Rule**, select a data load rule or click **Add**.
3. Select the **Source Options** tab.
4. Select the **Amount Type**:
 - Monetary
 - Statistical—The balance selection of entered or functional currency does not apply.
 - Monetary and Statistical
5. In **Signage Method**, select the method to flip the sign of amounts when data is loaded.

Available methods:

- Absolute—Loads the default debit or credit sign.

Table 3-13 Absolute Signage rules

Account Type	GL (GAAP)	EPM (Absolute)
Revenue	Naturally negative	Signage flipped
Liability	Naturally negative	Signage flipped
Equity	Naturally negative	Signage flipped
Expense	Naturally positive	Signage unchanged

Table 3-13 (Cont.) Absolute Signage rules

Account Type	GL (GAAP)	EPM (Absolute)
Asset	Naturally positive	Signage unchanged

- Same as source—Loads the same sign as recorded in the source system.
- Reverse from source—Loads the reverse of the sign as recorded in the source system.

6. Select the **Amount for Balance Sheet Accounts** and **Amount for Income Statement Accounts**:

- **YTD**—Year-to-date account balance, where account balances are accumulated from the beginning of the year to the current period. Typically, balance sheet accounts (assets, liabilities, and equities) are specified with a YTD balance.
- **Periodic**—Account balance for the specific period. Typically, income statement accounts (revenues and expenses) are specified with a periodic balance.

7. From **Currency Type**, select the currency type by which to extract balances:

- **Functional**—Balances stored in the base currency of the selected ledger or business unit (local currency)
- **Entered**—Balances associated with a currency attached to primary ledger other than the functional/local currency that you specify in the Currency Code field.

 Note:

The default when Oracle Hyperion Financial Data Quality Management, Enterprise Edition pulls from PeopleSoft is POSTED_TOTAL_AMT. If you select an entered currency, FDMEE pulls from POSTED_TRAN_AMT. If you want the BASE amount, then create an alternate schema and create a view PS_LEDGER to switch the amount.

8. From **Currency Code**, select the ISO 4217 currency code to use with an entered currency type.

For example, enter **EUR** to select the EURO currency code.

9. Select the **Ledger Group**.

10. Select the **Ledger**.

In PeopleSoft, a business unit may have multiple ledger groups. In this list, FDMEE displays only the ledger groups associated with the ledger.

11. **Optional:** To select book code values, click , select book code values, and then click **OK**.

To clear a book code, click . Then, in the Select Book Code dialog box, clear book codes, and then click **OK**.

12. To select budget values, click , select budget values, and then click **OK**.

You specify the budget values when the ledger that you selected has the data table "Ledger_Budg."

To deselect budget values, click . Then, in Select Budget Scenario values, clear values, and then click **OK**.

13. Click **Save**.

14. Define target filter options.

After you define target filter options, run the data rule. See [Running Data Load Rules](#).

Defining Source Filter Options for JD Edwards GL Source Systems

When defining data load mapping details, define the data to extract, including whether to extract the company code, ledger type, amount type.

To define the source filter options for a JD Edwards GL source system:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.

2. In **Data Load Rule**, select a data load rule or click **Add**.

3. Select the **Source Options** tab.

4. Select the **Amount Type**.

Select **PTD** for Period to Date, or **YTD** for Year to Date balances.

5. Select the **Company Code**.

Specify the company code in four characters or less using alphanumeric characters.

6. Select the **Ledger**.

Specify the ledger in two characters or less using alphanumeric characters from the JD Edwards source system. For example, ledger types include:

- AA—Actual
- BA—Budget
- CA—Original Currency Transaction

7. Click **Save**.

Defining Source Filter Options for SAP Adapters

This section provides background information on the SAP integration, and specific adapter detail.

SAP Integration Process Background

Oracle Data Integrator creates an ABAP program to extract the data from SAP into a file and then transfers the file to a FTP Server or Shared File system. The ABAP program is generated based on the mappings in the import format and options defined in the rule. The ABAP program is created and uploaded to SAP system when you run a data load rule. In the SAP deployment, ABAP programs are created and modified in a development environment, and locked from modification in the production environment.

In order to support this requirement, ODI provides an option `UPLOAD_ABAP_CODE` in the SAP Load Knowledge Module. This option is set to "Yes" in a development environment, and "No" in a production environment. The ABAP programs are transferred from development to production using SAP Transport requests.

Because the ABAP code that is run in a production environment is not modified, users should create the import format in the exact format as the development environment. In addition there are options in the data rule that impact the logic of the ABAP program. These options must be set to the same value as they are set in the development environment. Changing these options in the data rule in a production environment does not have any impact on the extraction process. If changes are required to these options, make them in the development environment and the ABAP code transported to the production environment. The options that impact the ABAP code generation are noted in the SAP adapter sections.

Defining Source Filter Options for the `SAP_FDM_GLBALANCES_CLASSICS` Adapter

Before executing a data load rule using a SAP source adapter, you are required to have generated the Import Format's ODI Scenario. Once the ODI Scenario exists in the Execution Repository, you can execute the data load rule any number of times.

To define the source filter options for the `SAP_FDM_GLBALANCES_CLASSICS` adapter:

1. In **Include Account Description**, select one of the following:
 - **Yes**—include the GL Account description
 - **No**—include all other conditionsImpacts ABAP program
2. In **Amount Type**, select one of the following:
 - **PTD**—Period to Date
 - **YTD**—Year to DateNo impact on ABAP program
3. In **Company Code**, specify the company code in four characters or less using alphanumeric characters.
No impact on ABAP program
4. In **Currency Type**, specify one of the following types:
 - **00**—Transaction currency
 - **10**—Company code currency
 - **30**—Group currencyNo Impact on ABAP program
5. In **Language**, specify the language code in two characters or less, using uppercase characters.
For example, specify "EN" for English.
Refer to the SAP documentation for the language code.
No impact on ABAP program
6. In **Ledger**, specify the ledger code in two characters or less using alphanumeric characters.

No impact on ABAP program

7. In **Record Type**, select one of the following record types:

- **0**—Actual
- **1**—Plan

Impacts ABAP program

Defining Source Filter Options for the SAP_FDM_GLBALANCES_NEW Adapter

Before executing a data load rule using a SAP source adapter, you are required to have generated the Import Format's ODI Scenario. Once the ODI Scenario exists in the Execution Repository, you can execute the data load rule any number of times.

To define the source filter options for a SAP_FDM_GLBALANCES_NEW adapter:

1. In **Amount Type**, select one of the following:

- **PTD**—Period to Date
- **YTD**—Year to Date

No impact on ABAP program

2. In **Company Code**, specify the company code in four characters or less using alphanumeric characters.

No impact on ABAP program

3. In **Currency Type**, select one of the following types:

- **00**—Transaction currency
- **10**—Company code currency
- **30**—Group currency
- **40**—Hard currency
- **50**—Index based currency
- **60**—Global company currency

No impact on ABAP program

4. In **Include Account Description**, select one of the following:

- **Yes**—include the GL Account description
- **No**—include all other conditions

Impacts ABAP program

5. In **Language**, specify the language code in two characters or less, using uppercase characters.

For example, specify "EN" for English.

Refer to the SAP documentation for the language code.

No impact on ABAP program

6. In **Ledger**, specify the ledger code in two characters or less using alphanumeric characters.

No impact on ABAP program

7. In **Record Type**, select one of the following record types:

- **0**—Actual
- **1**—Plan

No impact on ABAP program

Defining Source Filter Options for the SAP_FDM_COST_CENTER Adapter

Before executing a data load rule using a SAP source adapter, you are required to have generated the Import Format's ODI Scenario. When the ODI Scenario exists in the Execution Repository, you can execute the data load rule any number of times.

To define the source filter options for a SAP_FDM_COST_CENTER adapter:

1. In **Activity Type**, select one of the following:
 - **Yes**—include the activity type
 - **No**—exclude the activity typeImpacts ABAP program
2. In **Activity Type Description**, select whether to include or exclude the activity type description:
 - **Yes**—include the activity type description
 - **No**—exclude the activity type descriptionImpacts ABAP program
3. In **Amount Type**, select one of the following:
 - **PTD**—Period to date balances
 - **YTD**—Year to date balancesNo impact on ABAP program
4. In **Controlling Area**, specify the controlling area in four characters or less using alphanumeric characters.
No impact on ABAP program
5. In **Include Cost Element Description**, select whether to include the cost element description:
 - **Yes**—include the cost element description
 - **No**—exclude the cost element descriptionImpacts ABAP program
6. In **Currency Type**, select one of the following:
 - **20**—Controlling Area currency
 - **00**—Transaction currency
 - **70**—Cost Center currency
 - (blank)—leave blank when the **Quantity** filter is **Yes** or the **Activity Type** filter is **Yes**.Impacts ABAP program
7. In **Flow Check**, select one of the following:
 - **External**—load external balances

- **Internal**—load internal allocations
Impacts ABAP program
- 8. In **Language Code**, specify the language code in two characters or less, using uppercase characters.
For example, specify "EN" for English.
No impact on ABAP program
- 9. In **Ledger Code**, specify the ledger code in two characters or less using alphanumeric characters.
No impact on ABAP program
- 10. In **Statistical Key Figure**, select to extract statistical key figure totals:
 - **Yes**—extract the data for Statistical key figure totals
 - **No**—extract data for activity type totalsImpacts ABAP program
- 11. In **Group Code**, select one of the following group codes:
For external balances, select:
 - **0101**—data extracted is for the Cost Center Group
 - **0102**—data extracted is for the Account GroupFor internal balances, select:
 - **0101**—data extracted is for the Cost Center Group
 - **0102**—data extracted is for the Account Group
 - **0104**—data extracted is for the Statistical Key Figure Group
 - **0105**—data extracted is for the Account Group
 - (Null)—no grouping is requiredImpacts ABAP program
- 12. In **Value Type**, select one of the following:
 - **04**—Actual
 - **01**—PlanNo impact on ABAP program

Defining Source Filter Options for the SAP_FDM_PROFIT_CENTER Adapter

Before executing a data load rule using a SAP source adapter, you are required to have generated the Import Format's ODI Scenario. Once the ODI Scenario exists in the Execution Repository, you can execute the data load rule any number of times.

To define the source filter options for a SAP_FDM_PROFIT_CENTER adapter:

1. In **Amount Type**, select one of the following:
 - **PTD**—Period to date balances
 - **YTD**—Year to date balancesNo impact on ABAP program

2. In **Controlling Area**, specify the value for the controlling area.
No impact on ABAP program.
3. In **Currency Type**, select one of the following:
 - **10**—Company code currency
 - **00**—Transaction currency
 - **70**—Profit Center currency
 - (blank)—Select blank when **Quantity** is set to **YES**.No impact on ABAP program
4. In **Dummy Prctr**, select:
 - **Yes**—include balances related to the dummy profit center.
 - **No**—include other conditions.No impact on ABAP program
5. In **Language**, select the language code in two characters or less, using uppercase characters.
For example, specify "EN" for English.
No impact on ABAP program
6. In **Ledger**, select the ledger code in two characters or less using alphanumeric characters.
No impact on ABAP program
7. In **Statistical Key Figure**, select to extract statistical key figure totals:
 - **Yes**—extract the data for Statistical key figure totals
 - **No**—extract data for activity type totalsImpacts ABAP program
8. In **Group Code**, select one of the following group codes:
 - **0106**—Profit Center Group
 - **0109**—Account Group
 - (blank)—no group is requiredImpacts ABAP program
9. In **Record Type**, select one of the following types:
 - **0**—Actual
 - **1**—PlanNo impact on ABAP program

Defining Source Filter Options for the SAP_FDM_CUSTOMER_BALANCES Adapter

Before executing a data load rule using a SAP source adapter, you are required to have generated the Import Format's ODI Scenario. Once the ODI Scenario exists in the Execution Repository, you can execute the data load rule any number of times.

To define the source filter options for a SAP_FDM_CUSTOMER_BALANCES adapter:

1. In **Amount Type**, select one of the following:
 - **PTD**—Period to date balances
 - **YTD**—Year to date balances

No impact on ABAP program
2. In **Company Code**, specify the company code in four characters or less using alphanumeric characters.

No impact on ABAP program
3. In **Currency Type**, select one of the following:
 - **10**—Company code currency
 - **00**—Transaction currency

No impact on ABAP program
4. In **Customer**, select the customer code in ten characters when the balance is required for a specific customer.

Otherwise, set to blank.

No impact on ABAP program
5. In **Customer Details**, select to include customer details:
 - **Yes**—include customer details
 - **No**—exclude customer details

Impacts ABAP program
6. In **Flow Check**, select one of the following:
 - **Open**—load open item balances
 - **Cleared**—load cleared item balances
 - **All**—load all item balances

Impacts ABAP program
7. In **Special GL Transactions**, select one of the following:
 - **Yes**—load special General Ledger balances
 - **No**—load other balances

Impacts ABAP program
8. In **Trading Partner**, select to include trading partner balances:
 - **Yes**—load trading balances
 - **No**—load other conditions

No impact on ABAP program

Defining Source File Options for the SAP_FDM_VENDOR_BALANCES Adapter

Before executing a data load rule using a SAP source adapter, you are required to have generated the Import Format's ODI Scenario. Once the ODI Scenario exists in the Execution Repository, you can execute the data load rule any number of times.

To define the source filter options for the SAP_FDM_VENDOR_BALANCES adapter:

1. In **Amount Type**, select one of the following:
 - **PTD**—Period to date balances
 - **YTD**—Year to date balancesNo impact on ABAP program
2. In **Company Code**, specify the company code in four characters or less using alphanumeric characters.
No impact on ABAP program
3. In **Currency Type**, select one of the following:
 - **10**—Company code currency
 - **00**—Transaction currencyNo impact on ABAP program
4. In **Flow Check**, select one of the following:
 - **Open**—load open item balances
 - **Cleared**—load cleared item balances
 - **All**—load all item balancesImpacts ABAP program
5. In **Special GL Transactions**, select one of the following:
 - **Yes**—load special General Ledger balances
 - **No**—load other balancesImpacts ABAP program
6. In **Trading Partner**, select to include trading partner balances:
 - **Yes**—load trading balances
 - **No**—load other conditionsNo impact on ABAP program
7. In **Vendor**, specify the vendor code in ten characters when the balance is required for a specific vendor.
Otherwise, set to blank.
No new ABAP code is uploaded for this filter.
8. Specify **Vendor Details**, select whether to include vendor details:
 - **Yes**—include vendor details
 - **No**—exclude vendor detailsNo impact on ABAP program

Defining Source Filter Options for the Open Interface Adapter

Before executing a Data Rule using open interface adapter, you are required to have generated the Import Format's ODI Scenario. Once the ODI Scenario exists in the Execution Repository, you can execute the Data Rule any number of times.

To define the source filter options for an open interface adapter:

1. In **Batch Name**, enter the name of the batch used to identify the data in the interface table.
2. In **Record Type**, specify whether to delete data after importing the data by selecting **Y** (for yes) or **N**.
To delete the data, select **Y**. To retain the data, select **N**.
3. Click **Save**.

Defining Source Filter Options for Financial Management

When defining data load rule details, you can specify various data extraction options.

For journals, the journals are extracted based on Scenario, Year, Period, Entity and Value.

For data (including journalized value members), any dimension can be filtered by any member or multiple members.

You can select an individual member. If you do not select specific members, the system assumes that you want to extract all members for the dimension. However, if you specifically select members, the system displays a plus sign (+) next to the dimension to indicate multiple selections.

To define the source filter options for source systems:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. In **Data Load Rule**, select a data load rule or click **Add**.
3. Select the **Source Options** tab.
4. Click **Add**.
5. In the **Source Filters** area, click **Add**.
6. Select the **Dimension Name**.
7. To assign a filter condition, perform an action:
Enter a member name or filter condition in the Filter Condition text box.

To assign a filter condition, perform an action:

- Enter a member name or filter condition in the Filter Condition text box.

For example, enter a member name or filter condition using Oracle Essbase syntax. Depending on the dimension, you can select one or more members as a filter condition used for extracting the budget data. For example, for the Entity dimension, you may select the following members: E1, E5, and E6.

- Click  to display the Member Select screen and select a member using the member selector. Then, click **OK**.

The Member Selector dialog box is displayed. The member selector enables you to view and select members within a dimension. Expand and collapse members within a dimension using the [+] and [-].

The Selector dialog box has two panes—all members in the dimension on the left and selections on the right. The left pane, showing all members available in the dimension, displays the member name and a short description, if available. The right pane, showing selections, displays the member name and the selection type.

You can use the V button above each pane to change the columns in the member selector.

You can also click **Refresh Members** to show the latest member list.

 Note:

Assign filters for dimensions. If you do not assign filters, numbers from the summary members are also retrieved.

To use the member selector:

In the list of available dimensions and members on the left, select a member and click . To deselect a member from the list of members, click .

To add special options for the member, click  and select an option.

In the member options, "I" indicates inclusive. For example, "IChildren" adds all children for the member, including the selected member, and "IDescendants" adds all the descendants including the selected member. If you select "Children," the selected member is not included and only its children are included.

The member is moved to the right and displays the option you selected in the Selection Type column. For example, "Descendants" displays in the Selection Type column.

 Tip:

To clear all members from the list of selections, click .

Click **OK** twice to continue defining the source filter details.

The selected member is displayed in Essbase syntax in the Filter Condition field.

Click **Save**.

Defining Data Load Rule Details for a File-Based Source System

When defining data load detail for a file-based data load system, load data to a single period or a range of periods. For a single period, enter the file name in the data rule, and then run the rule for a single period. To load multiple periods, create a file for each period and append the period name or period key to the file name. When the rule is executed for a range of periods, the process constructs the file name for each period, and then uploads the appropriate data to the POV.

To define the data load details for a file-based source system:

1. In **Name**, enter the data load rule name.
2. In **Category**, select a category.

The categories listed are those that you created in the Oracle Hyperion Financial Data Quality Management, Enterprise Edition setup, such as "Actual." See [Defining Category Mappings](#).

3. Optional: In **Description**, specify a description of the data load rule.
4. Optional: From the **Target Plan Type** drop, select the plan type of the target system.
5. **Optional:** In **Import Format**, if the file type is a multiple period *text* file (with contiguous periods, or noncontiguous periods), select the import format to use with the file, so you can override the import format. For example, specify an import format for single and multiple period data rules, which enables you to load single or multiple period files from the same location. In this case, the import format selected must have the same target as the location selected in the POV. If the import format is unspecified, then the import format from the location is used.

The starting and ending period selected for the rule determine the specific periods in the file when loading a multiple period text file.

In the file, when amounts are unavailable for contiguous periods, then you can explicitly map the respective amount columns to required periods in the data rule in Data Load Mapping. When you execute the rule, the data is loaded to the periods as specified in the explicit mapping.

6. **Optional:** Enter a description.
7. In **Directory**, enter the relative path where the file is located.
8. Select the **Source Options** tab.
9. In the **File Name** field, enter the static name of the file.

When only the file name is provided, then data must be entered for a single period on the Rules Execution window.

To load multiple periods, create a file for each period and append a period name or period key to the file name. When the rule is executed for a range of periods, the process constructs the file name for each period and uploads it to the appropriate POV.

**Note:**

If used in a batch script, period names cannot include spaces.

To navigate to a file located in a FDMEE directory, click **Select**, and then choose a file on the **Select** screen. You can also select **Upload** on the **Select** screen, and navigate to a file on the **Select a file to upload** screen.

If you do not specify a file name, then FDMEE prompts you for the file name when you execute the rule.

10. To load data into multiple periods, in the **File Name Suffix Type** drop-down, select **Period Description** or **Period Key**.

A suffix is appended to the file name, and FDMEE adds the file extension after adding the suffix. If you leave the file name blank, then FDMEE looks for a file with Suffix. When the file name suffix type is provided, then the file name is optional in this case, and it is not required on the Rule Execution window.

If the file name suffix type is a period key, the suffix indicator and period date format are required (as the suffix set) in the file name, and must be validated as a valid date format.

For example, specify:

- a. 1_Jan-2013.txt
- b. 1_Feb-2013.txt
- c. 1_Mar-2013.txt

In this case, when you run the rule, enter 1_.txt in the file name field and select "Period Name" for the suffix indicator. Then run the rule for the January to March periods.

11. In **Period Key Date Format**, specify the data format of the period key that is appended to the file name in JAVA date format (SimpleDateFormat).
12. Click **Save**.

Defining Source Parameters for Planning and Essbase

In data synchronization, when Oracle Hyperion Planning and Oracle Essbase are the source systems, you can specify additional source parameters.

To define source options:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. In **Data Load Rule**, select a data load rule for a Planning and Essbase source, and then click **Add**.
3. Select the **Source Parameters** tab.
4. In **Extract Dynamic Calculated Data**, specify to include dynamically calculated data.

Note:

If you set the **Extract Dynamic Calculated Data** option on the Data Load Rule screen to "Yes," and a leaf level member's (Level 0) Data Storage is set to "Dynamic," then the data is not picked up by the extraction process. To pick up the data, set the member's Data Storage to something besides "Dynamic," to include the value in the selection from the source application.

- **Yes**—Dynamically calculated values are included in the export.
By default, dynamically calculated data is included.
 - **No**—Dynamically calculated values are excluded in the export.
5. In **Data Precision**, specify the number of decimal places displayed in numbers to be exported.

Data precision refers to numeric data with the emphasis on precision (accuracy). Depending on the size of a data value and number of decimal positions, some numeric fields may be written in exponential format; for example, 678123e+008. You might consider using data precision when data ranges from very large to very small values. The output files typically are smaller and data values are more accurate.

The default value for this option is 16.

6. In **Data Number of Decimal**, specify the maximum number of decimal positions to be exported.

Specify a value between 0 and 16. If no value is provided, the number of decimal positions of the data to be exported is used, up to 16 positions, or a value determined by Data Precision option if that value is specified.

This parameter is used with an emphasis on legibility; output data is in straight text format. Regardless of the number of decimal positions in the data, the specified number is output. Note that it is possible the data can lose accuracy, particularly if the data ranges are from very large values to very small values, above and below the decimal point.

By default, 16 positions for numeric data are supported, including decimal positions. If both the Data Precision option and the Data Number of Decimal option are specified, the Data Precision option is ignored.

7. Click **Save**.

Defining Source Parameters for Financial Management

When Oracle Hyperion Financial Management is the source system, you can specify additional source parameters such as the journal status, journal type, balance type, and the type of data to extract.

To define source parameter options:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. In **Data Load Rule**, select a data load rule.
3. Go to the **Source Parameters** region.
4. Click **Add**.
5. Select the parameter(s) row to add, and then select the parameter value from the drop-down:
Options:
 - **Add Journal Status:** Specify the journal status to extract.
The journal status indicates the current state of the journal. The status of a journal changes when you create, submit, approve, reject, or post the journal.
Options:
 - Working—Journal is created. It has been saved, but it can be incomplete. For example, it might need to have a label or single entity assigned.
 - Submitted—Journal is submitted for approval.
 - Approved—Journal is approved for posting.
 - Rejected—Journal is rejected or unposted.
 - Posted—Journal adjustments are posted to the database.
 - **Add Journal Type:** Specify the type of journal:
 - Auto-Reversal—Loads a posted auto-reversing journal.
 - Auto-Reversing—Loads an auto-reversing journal that contains adjustments that need to be reversed in the next period. That is, the journal posts in the next period by reversing the debit and credit.

- Regular—Load journals using the Replace mode, which clears all data for a particular journal label before loading the new journal data.
- **Add Balance Type:** Specify the type of balance to extract:
 - Balanced—All debits and credits are balanced.
 - Unbalanced—Debits and Credits are not balanced.
 - Balanced by entity—Debits and Credits for particular entity are balanced.
- From **Extract Type**, select one of the following:
 - Data—Extract base-level input data and some calculated data from an application. When you extract data, you must specify a member for the Scenario and Year dimensions. You can specify one or more members for the Period, Entity, and Account dimensions.
 - Journal—You can extract base-level input data and some calculated data from an application. When you extract data, you must specify a member for the Scenario and Year dimensions. You can specify one or more members for the Period, Entity, and Account dimensions.
- In **Extract Dynamic Account**, specify to extract dynamic accounts. Dynamic accounts are accounts with values that are dynamically calculated when the data is requested. The values for dynamic accounts are not stored. The most common type of dynamic calculation is ratio calculation.

Extract Dynamic Account options:

- **Yes**—Dynamic accounts are extracted.
- **No**—Dynamic accounts are not extracted.

By default, dynamic accounts are not extracted.

- In **Extract Calculated Data**, specify whether to extract calculated data. Options:
 - Consolidated data for parent entities
 - Base accounts and custom dimension members that are calculated by rules
 - Parent accounts intersecting with their respective CustomTop member and [ICPTop] member. If the CustomTop metadata attribute is blank or ALL, then the [None] member is used.

Extract calculated data options:

- **Yes**—Calculated data is extracted.
- **No**—Calculated data is not extracted.

By default, calculated data is not extracted.

- In **Extract Derived Data**, specify whether to extract derived data. Derived data is data derived from other data values, such as calculated data or arithmetic formulas.
 - **Yes**—Derived data is extracted.
 - **No**—Derived data is not extracted.

6. Click **Save**.

Managing Data Load Rules

You can perform the following tasks:

- Edit data load rules—See [Editing Data Load Rules](#).
- Run data load rules—See [Running Data Load Rules](#).
- Delete data load rules—See [Deleting Data Load Rules](#).
- View data load rules before executing them—See [Using the Data Load Workbench](#).
- Schedule data load rules—[Scheduling Data Load Rules](#)
- Check the data rule process details—See [Viewing Process Details](#).

Editing Data Load Rules

If the data load rule is not in the process of running, you can modify rule details.

To edit data rules:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. Select the data load rule.
3. Modify any of the data load rule details, as necessary.
4. Click **Save**.

Running Data Load Rules

You run the data load rule to load updates and push the data into the target application. All submitted rules are processed by Oracle Data Integrator. When you submit a data load rule, specify the data extract options.

Data Load Rules can be executed by selecting one of the methods below:

- Execute command on the Data Load Rule screen.
- Import Source option in the Data Load Workbench option.
- Executing a batch. See [Executing Batches](#).
- Running a batch script. See [Working with Batch Scripts](#).

When a data load rule is run, Oracle Hyperion Financial Data Quality Management, Enterprise Edition loads the data, and creates a drill region (optional), which enables users to drill through to the source data.

Note:

In Oracle Financial Consolidation and Close Cloud for YTD data loads, data is stored in Periodic view. In this case, the user must select this option so that a “pre-processing” is done to convert the YTD data from the file to periodic data for loading purpose.

When you run a data load rule, you have several options:

**Note:**

When a data load rule is run for multiple periods, export step occurs only once for all of the period.

- **Import from Source**—FDMEE imports the data from the source system, performs the necessary transformations, and exports the data to the FDMEE staging table.

Select this option only when:

- You are running a data load rule for the first time.
- Your data in the source system changed. For example, if you reviewed the data in the staging table after the export, and it was necessary to modify data in the source system.

In many cases, source system data may not change after you import the data from the source the first time. In this case, it is not necessary to keep importing the data if it has not changed.

When the source system data has changed, you need to recalculate the data.

**Note:**

Oracle E-Business Suite and source imports require a full refresh of data load rules. The refresh only needs to be done once per chart of accounts.

- **Export to Target**—Exports the data to the target application.

Select this option after you have reviewed the data in the staging table and you want to export it to the target application.

**Note:**

Select both options only when the data has changed in the source system *and* to export the data directly to the target application.

To submit the data load rule for an Oracle Hyperion Planning General Ledger or Enterprise Resource Planning (ERP):

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. In **Data Load**, select the data load rule.
3. Click **Execute**.

When the data rule is run for Oracle Hyperion Financial Management target applications, the Exchange Rates from the source are populated only up to the FDMEE interface table AIF_HS_EXCHANGE_RATES. The Core Exchange Rates table in Financial Management is not updated.

4. From **Execute Rule**, to extract data, metadata or both from the source system and push it into target applications, select **Import from Source**, and then select the **Start Period** and **End Period**.

**Tip:**

You can use a utility outside of FDMEE to view the data in the staging table. After you review the exported data, return to FDMEE, make modifications, and run the rule again. If you are sure that the information in the staging table is correct, run the rule again and select “Export to Target.”

5. Select **Recalculate** to remap all imported source data using the current mapping table and to recreate all logic accounts.
6. Select **Export to Target** to export data to the target application.
7. Select **Execute Check** to generate the date, and then run the Check Report.
8. In **Start Period**, select the beginning period of the POV from which to import the data from the source system.
9. In **End Period**, select the ending period of the POV to which to import the data from the source system.
10. In **Import Mode**, select the mode to extract data all at once for an entire period or incrementally during the period.

**Note:**

The snapshot import mode is the only way data can be extracted from a SAP and JD Edwards source system.

Data extract types:

- **Snapshot**—Extracts everything for the selected source set for an entire period
 - When source data for the selected period has never been run, FDMEE extracts the data from the source.
 - When the source data for the selected period has been run, FDMEE extracts the data from the FDMEE staging tables and not from the source.

When you have locations that extract from the same Enterprise Resource Planning (ERP) source, FDMEE extracts the data once. When you load data to Financial Management from the E-Business Suite for a selected period, and then run the integration to ARM for the same source and period, FDMEE does not pull again from E-Business Suite, but uses the data in the interface tables. This results in a significant performance gain for any subsequent data loads. The first extraction takes the longest, but any other subsequent extractions are faster.
- **Incremental**—Extracts the records that were added after the previous data extract
- **Full Refresh**—Performs a clean extraction from the source system, thereby clearing any existing data rows in the appropriate FDMEE staging tables for a given source Ledger (or Business Unit) and source period.

 Note:

The import mode options (Snapshot, Incremental, and Full Refresh) apply only to data rules in a location using a standard import format.

 Note:

E-Business Suite and Fusion source imports require a full refresh of data load rules before export after upgrading from an 11.1.2.2 release.

 Note:

If you run a data load in Full Refresh mode in the Account Reconciliation Manager (ARM), select all locations that have data. Otherwise, FDMEE contains data for locations not selected in ARM (but ARM does not). This results in a discrepancy between what is in FDMEE and what is in ARM.

 Note:

E-Business Suite and Fusion source imports require a full refresh of data load rules before export after upgrading from an 11.1.2.2 release.

11. In **Export Mode**, select the mode for exporting data:

- **Store Data**—Inserts the data from the source or file into the target application, replacing any current value.
- **Replace Data**—Replaces the data in the application with the data in the load file. The Replace Data option clears account values only for data between the start-period and end-period parameters specified when running the data rule. When account values are cleared, the data from the data load file is then pushed into the application.

 Note:

You can clear an Essbase ASO cube before you load the data by selecting the replace option

- **Add Data**—Adds the value from the source or file to the value that exists in the target application. For example, if you have 100 in the source, and 200 in the target, then the result is 300.
- **Subtract Data**—Subtracts the value in the source or file from the value that exists in the target application. For example, if you have 300 in the target, and 100 in the source, then the result is 200.

- **Override All Data**—Clears all data in the target, and then loads from the source or file. For example, if you have a year of data in your Planning application, but are loading only a single month, this option clears the entire year before performing the load.

When you select to override all data, the following message is displayed "Warning: Override All Data option will clear data for the entire application. This is not limited to the current Point of View. Do really want to perform this action."

Available export modes for Financial Management:

- **Merge**—Overwrites the data in the application with the data in the load file. For each unique point of view that exists in the data file and in the application, the value in the data file overwrites the data in the application.

**Note:**

If the data load file includes multiple values in the file for the same point of view, the system loads the value for the last entry.

**Note:**

Data in the application that is not changed by the data load file remains in the application.

- **Accumulate**—accumulate the data in the application with the data in the load file. For each unique point of view in the data file, the value from the load file is added to the value in the application.
- **Replace**—Replaces the data in the application with the data in the load file. For each unique combination of Scenario, Year, Period, Entity, and Value in the data file, the Replace option clears all account values from the application, then loads the value from the data file.

**Note:**

You can clear an Essbase ASO cube before you load the data by selecting the replace option

- **Replace by Security**—Performs a data load in Replace mode in which only the members to which you have access are loaded. This option enables you to perform a data load in Replace mode even if you do not have access to all accounts. When you perform the Clear operation for a period in a sub-cube, only the cells to which you have access are cleared. Data, cell text, and line item detail are cleared, but cell attachments are not cleared.

12. Select **Include Exchange Rates** to load exchange rates.

13. Click **Run**.

After you click Run, the rule is locked from any updates to ensure that the drill through path is intact. To check the status of the rule, see [Checking the Data Load Rule Status](#).

Scheduling Data Load Rules

The scheduling jobs feature provides a method to orchestrate the execution times of data load rules.

To schedule data load rules to run:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. In **Data Load**, select the data load rule.
3. Click **Schedule**.

For information on scheduling jobs, see [Scheduling Jobs](#).

To cancel a scheduled job:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. In **Data Load**, select the data load rule.
3. Click **Cancel Schedule**.

When you cancel a job from the Oracle Hyperion Financial Data Quality Management, Enterprise Edition user interface, all instances of a schedule for the object selected are cancelled. To cancel a specific instance of a schedule, cancel the job from the ODI studio or ODI console.

Checking the Data Load Rule Status

After you run a data rule, you can check the status on the Process Details page. See [Viewing Process Details](#).

Deleting Data Load Rules

You can delete data load rules created in Oracle Hyperion Financial Data Quality Management, Enterprise Edition. You cannot delete data load rules when they are running.

When you delete a rule, all data loaded using the data rule are also deleted.



Note:

After you delete data load rules, you can delete a source system. After you execute a deletion, users cannot drill through to an Enterprise Resource Planning (ERP) source.

To delete a data load rule:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. Enter the **Location Name** or click  to select the location.

3. Select the data load rule.
4. Click **Delete**.

Working with Target Options

When working with data load rules, you can specify target application options specific to a location. For example, using the Target Options feature, you can specify different data protection values for each location.

Note:

For information on the required target options for data load rules to write back, see [Defining Application Options for Essbase and Planning](#).

To specify integration information:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. From the **POV** bar, select a location.
3. Select the **Target Options** tab.
4. Add or modify any options.
See [Registering Target Applications](#).
5. Click **Save**.

Creating Custom Options

You can specify free form text or a value about a location or data load using the integration option feature. Text or values entered can be used with your Oracle Hyperion Financial Data Quality Management, Enterprise Edition scripts.

Additionally, if you use Oracle Data Relationship Management to export dimensions and hierarchies from ERP systems to Data Relationship Management, or import data load mapping from Data Relationship Management to FDMEE, you can specify Data Relationship Management import and export profiles.

Data Relationship Management integration is enabled on the Target Application screen. See [Registering Target Applications](#) for more information.

To specify integration information:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. Select the **Custom Options** tab.
3. In **Integration Option 1-4**, specify the free form text or value, and click **OK**.
The information that you specify is accessible from the Integration Option fields of the Location table.

Loading Exchange Rates to Financial Management

When you define a data rule for an Oracle Hyperion Financial Management target application, you can specify how to extract exchange rates from the Enterprise

Resource Planning (ERP) source system. Exchange rates are captured, if applicable, for the currency rate account for the Beginning Rate, currency rate account for the Ending Rate, and the currency rate account for the Average Rate.

If your target application has the multi-currency option enabled, you can specify how to process exchange rates. All rates are extracted and inserted into the AIF_HS_EXCHANGE_RATES table. This table is populated using the ISO currency code for each currency from the source system. The ISO numeric code is not used in this processing. Exchange rates are pushed into Financial Management based on a match between the ISO currency code in the AIF_HS_EXCHANGE_RATES table and the currencies defined in the multi-currency Financial Management application. (It is important to set up the currencies in the Financial Management application with the ISO currency).

To load exchange rates to Financial Management:

1. On the **Setup** tab, under **Register**, select **Target Application**.
2. In the **Target Application** summary grid, select a Financial Management target application.
3. After defining the application details in **Application Detail**, select the **Application Options** tab.
4. Specify the account names in the following fields:
 - Currency Rate Account for Beginning Rate
 - Currency Rate Account for Beginning Rate
 - Currency Rate Account for Beginning Rate

For information about working with other application options, see [Registering Target Applications](#).

Property Name	Value	Select
Translation Level	<Parent Currency>	
Enable Zero Loading	No	
Journal Status	Posted	
Journal Default Value	<Entity Curr Adjs>	
Journal Enable JV ID per Entity	Yes	
Journal Balancing Attribute	Unbalanced	
Currency Rate Account for Beginning Rate	Beginning Rate	
Currency Rate Account for Ending Rate	Ending Rate	
Currency Rate Account for Average Rate	Average Rate	
Currency Rate Entity		
Global User for Application Access		
Source Language for Member Description	AMERICAN	

5. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
6. In **Data Load Rule**, select a data load rule or click **Add**.
7. Select the **Source Options** tab.
8. Select the **Beginning**, **Ending**, and **Average** exchange rate options.

Enterprise Resource Planning (ERP) source systems maintain comprehensive exchange rate information for transaction processing. Target applications can use

this information by extracting the exchange rates. You can select a beginning, ending, and average rate type from the source system. (The types in the source system may not explicitly define those rates types but are mapped to the rates types in the Oracle Hyperion Financial Data Quality Management, Enterprise Edition interface table.)

For more information about source options, see [Defining Source Filter Options for E-Business Suite Source Systems](#).

Using Drilling Through

Oracle Hyperion Financial Data Quality Management, Enterprise Edition provides a framework that enables you to drill through from your Oracle Enterprise Performance Management System application to the onpremise application. Use drill through when you want to understand the source of a data value, or you need a granular level of detail for a value without leaving the workspace.

When you load data to an EPM System target application using FDMEE, you can specify an optional drill region flag. When set to "Yes," a drill region definition is passed to the target application, in addition to the actual data, to indicate that a cell is "drillable."

When drilling through from the EPM System application, a landing page is displayed in a separate workspace tab that shows all rows that comprise the amount from the selected cell in the Oracle Enterprise Performance Management Cloud application. From this landing page, you can open the source document or continue to drill-through to the defined source system landing page.

Drill through based on a URL requires that you are connected to the server on which the data resides. Drill through works only for data loaded through FDMEE. In addition because drill through is available in the target application, data load mappings must have at least one explicit mapping for the drill through to work.

Creating the Drill Region

A drill region is a named region of data intersections in an Oracle Enterprise Performance Management System application that can be viewed with the drill-through functionality. The data in the drill region is loaded to the EPM System application with Oracle Hyperion Financial Data Quality Management, Enterprise Edition. In the workspace, you can view the drill region in the data forms of the application.

The drill region stores the uniform resource locator (URL) to get back to FDMEE, and a region name. When you enable drill regions, FDMEE populates them in the FDMEE target application after data is loaded and consolidated. A cell is considered drillable in the target application when it is contained in the drill regions. For EPM System applications, the drill region includes the Entity, Account, Scenario, Year, and Period dimensions.

FDMEE creates drill region by scenarios. For any cube (Oracle Hyperion Planning plan types or Planning databases), the name of the drill region is FDMEE_<name of the scenario member>. When creating the drill region, FDMEE checks if a dimension is enabled for the drill.

Drill regions to the source system work only when there is an identical intersection of the data loaded by FDMEE. For example, if you loaded \$5000 to Entity2;Account2

intersection, you cannot drill-through from Entity2;ParentAccount2 since no data was loaded for that particular intersection.

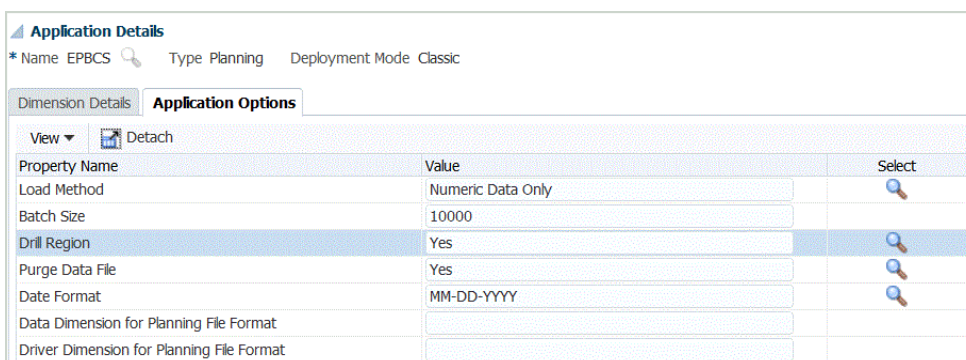
Members of enabled dimensions selected in data loads, are included in the drill region filter. If no dimensions are enabled, the following dimensions are enabled by default: Scenario, Version, Year, and Period. You can enable additional dimensions, and the subsequent data load considers members of newly enabled dimensions. If you disable any dimensions which were previously included in a drill region used for drill creation, members of such dimensions are not deleted during the subsequent data loads. If needed, you can remove obsolete members manually.

To add a drill region for the FDMEE target application:

1. On the **Setup** tab, under **Register**, select **Target Application**.
2. In the **Target Application** summary grid, select the EPM System target application.
3. Select the **Application Options** tab.
4. In **Drill Region**, enter **Yes**.

Note:

Administrators can set the drill region setting at the application level in the Target Application option. Additionally they can change the setting for a specific target application in data load rules.



Property Name	Value	Select
Load Method	Numeric Data Only	
Batch Size	10000	
Drill Region	Yes	
Purge Data File	Yes	
Date Format	MM-DD-YYYY	
Data Dimension for Planning File Format		
Driver Dimension for Planning File Format		

5. Click **Save**.

Drill Through Components

The Oracle Hyperion Financial Data Quality Management, Enterprise Edition landing page is called by a drill through URL. Two components define the drill-through URL:

- Server component—defined for each source system
- Detail Component – defined manually for file-based integrations

Adding the Server Component for the Drill Through URL

Use these steps to define the server component drill through URL in Oracle Hyperion Financial Data Quality Management, Enterprise Edition. The server component is added to the source system definition.

 Note:

To add a drill through URL:

1. On the **Setup** tab, under **Register**, select **Source System**.
2. Select the file to which to add the drill through.
3. In **Drill Through URL**, click .
4. In **Drill Through URL** edit window, specify the URL that identifies the URL to use for drilling through.

The URL to use needs to include the request-response method, protocol, and the resource name.

The protocol identifier indicates the name of the protocol used to fetch the resource. The Hypertext Transfer Protocol (HTTP) typically serves up the hypertext documents. HTTP is just one of many different protocols used to access different types of resources. Other protocols include a file.

Available request-response between a client and server methods are:

- GET—Form data is encoded into the URL. For example, specify: GET@http://www.oracle.com/. If no method is specified, then GET is the assumed request-response.
- POST—Form data is displayed in the message body. For example, specify: POST@http://www.oracle.com/.

The resource name is the complete address to the resource. The format of the resource name depends entirely on the protocol used, but for many protocols, including HTTP, the resource name contains one or more of the following components:

- Host Name—Specify the name of the machine on which the resource lives.
- File Name—The path name to the file on the machine.
- Port Name—The port number to which to connect (typically optional).

When specifying the resource information, use this format: http://<SERVER>:<PORT>

This URL identifies the server component of the drill through URL. In the import format, you include the detail component.

5. Click **OK** and then click **Save**.



Adding the Detail Component for the Drill Through URL

The detail component defines the parameters values of the drill through URL. Values not hard coded in the drill through URL can be referenced from the TDATESEG table using references to the table column names, or to the column names specified in the

import format. Drill through attributes that are different for each row of data, like the CCID (code combination ID for Oracle E-Business Suite applications) must be included in the input file and mapped to an attribute column in the import format. When formatting the drill URL, these parameters can be referenced from the TDATESEG table using the \$<TDATESEG_COLUMN>\$ notation, or from the import format using the \$\$<IMPORT_FORMAT_COL>\$\$ notation.

To add the detail component of the drill through URL:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. From the **Import Format Summary** section, select the import format.
3. In the **Import Format Detail** section, in the **Drill URL**, click .
4. Click **Add**, and specify the import format name, source, target, file type, and file delimiter.
5. In the **Drill URL** edit window, specify the parameters of the drill through URL.

Parameters may include any specific source system required information, and the attributes used to populate the actual drill region.

The attribute refers to the source columns (dimensions) in the TDATESEG table. You can provide the location of the attribute in the input file by specifying the starting location and length.

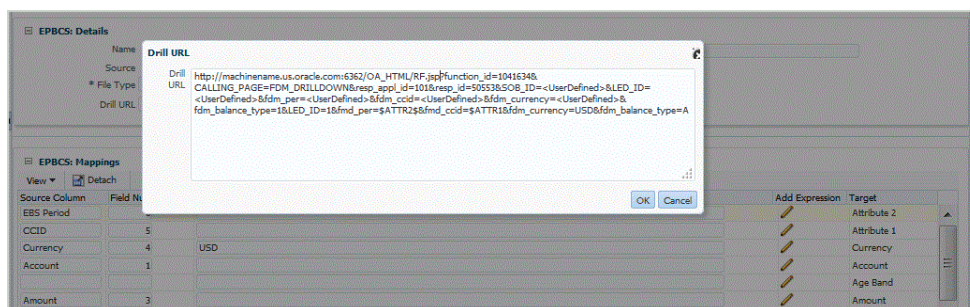
In the following example, the drill through URL enables a user to drill through to an E-Business Suite source system.

In addition to the attributes selected for the drill through, the function ID, calling page, and Resp_Appl_ID and Resp_id information are specified.

The function ID is E-Business Suite system specific, and is generated at the time the E-Business Suite system is installed and configured. The calling page parameter indicates the origin of the HTTP call. This is hard coded in the URL as **FDM_DRILLDOWN**. The Resp_Appl_ID and Resp_id parameters are the application ID and responsibility ID from E-Business Suite. These can be set to -1 if unknown. The E-Business Suite source system prompts the user for the user name, password and responsibility if not provided on initial authentication.

Attributes may include:

- SOB_ID—Internal Set of Books ID
- LED_ID—Internal Ledger ID
- fdm_per—the E-Business Suite period name. This value must be the same as the period name in the E-Business Suite system.
- fdm_ccid—the Code Combination ID (CCID) for the requested account.
- fdm_currency—the E-Business Suite ledger currency
- fdm_balance_type—"A" for Actual, "B" for Budget



6. In the Mappings grid of the import format, map the columns in the source column to the dimensions in the target application to which to drill through.
7. Click **OK** and then click **Save**.

Viewing the Drill Through Results

When you set up a drill through for data was loaded by Oracle Hyperion Financial Data Quality Management, Enterprise Edition, you can select the data form in the Oracle Enterprise Performance Management System application and then display the general ledger accounts and the hyperlinked balances that were used to populate the cells in the onpremise application.

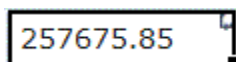
When you set up a drill through for data was loaded by FDMEE, you can select the data form in the Oracle Enterprise Performance Management Cloud application and then display the general ledger accounts and the hyperlinked balances that were used to populate the cells in the onpremise application.

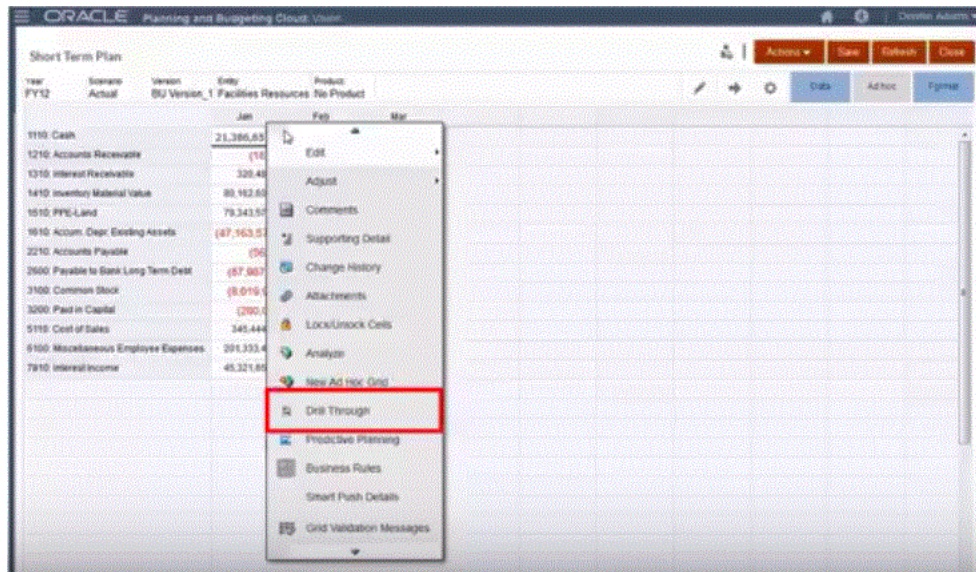
From a linked data value, you can drill through to the source system from the landing page and then view the associated journal entries and sub-ledger entries for the selected account.

To view the drill through results in the onpremise application

1. In the EPM Cloud application, open the data form of the loaded source data.
2. From a cell that contains drill-down data, click **Actions**, and then select **Drill Through**.

A drill through icon in the right top corner of a cell indicates that it has drill-through data:

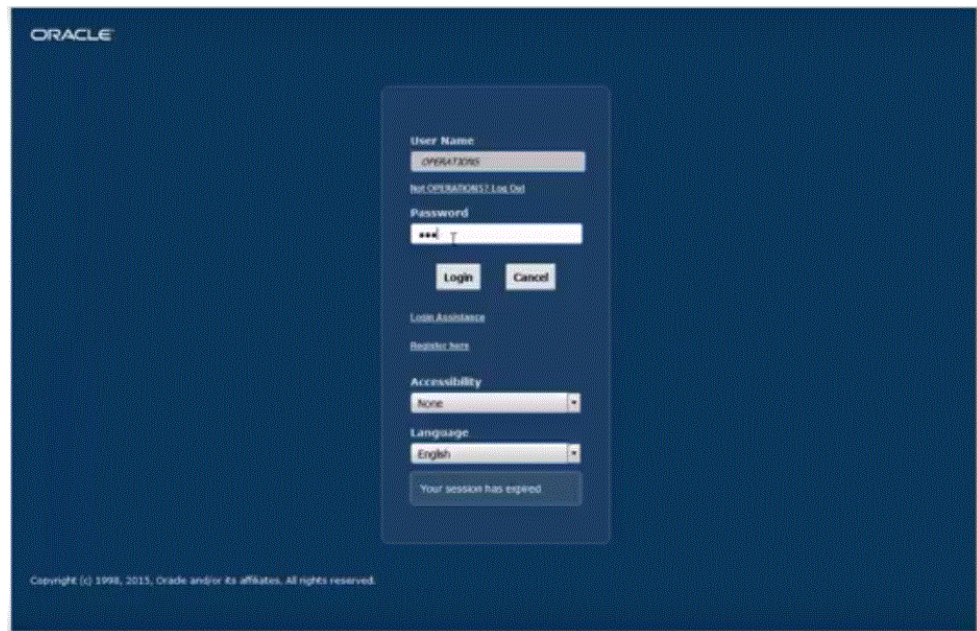




3. Click **Drill Through to Source**.
4. From the landing page, click the drill through icon.

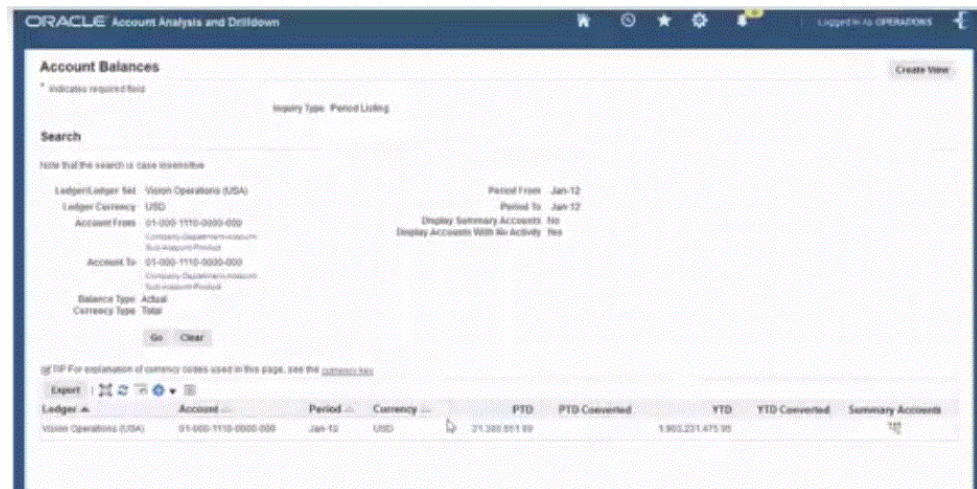


5. Click **Drill Through to Source**.
 6. Log into the onpremise application by providing any required credentials.
- In the following example, an Oracle E-Business Suite logon screen is shown.



7. Drill down to the level of detail that you want to view.

In the following example, you can drill through to the ledger details that support the balance:



In the next example, you can drill through to the sub-ledger that supports the balance:

Account Balances > Journal Lines: 01-000-1110-0000-000, Jan-12

Ledger: Vision Operations (USA)
Ledger Currency: USD

Account Description: Operations-No Department-Cash-No Sub Account-No Product
Balance Type: Actual

Search

Specify parameters and values to filter the data that is displayed in your results set.

☒ Show table data when all conditions are met.
☐ Show table data when any condition is met.

Batch is
Journal is
Line is
Source is

TIP For explanation of currency codes used in this page, see the [currency page](#).

Batch	Journal	Line	Source	Category	Currency	Entered Debit	Entered Credit	Accounted Debit	Accounted Credit
Receivables A 4704812 8805107	Jan-12 Receipts CAD	1	Receivables	Receipts	CAD	532,624.29		637,186.73	
Receivables A 4704812 8805107	Jan-12 Receipts EUR	1	Receivables	Receipts	EUR	600.00		77,862.97	
Receivables A 4704812 8805107	Jan-12 Receipts USD	1	Receivables	Receipts	USD	20,671,572.29		20,671,572.29	

[Return to Account Balances](#)

In the next example, you can view additional information associated with the balance:

Account Balances > Journal Lines > View Journal Entry Lines

Ledger: Vision Operations (USA)
Period: Jan-12
Currency: USD
Account: 01-000-1110-0000-000

GL Batch Name: Receivables A 4704812 8805107
GL Journal Entry Name: Jan-12 Receipts CAD
Accounted DR: 637186.73
Accounted CR:

Advanced Search

Specify parameters and values to filter the data that is displayed in your results set.

☒ Show table data when all conditions are met.
☐ Show table data when any condition is met.

Balancing Segment is
Natural Account Segment is
GL Date is

Select Subledger Journal Entry Line:

Select Subledger Journal Entry Line	Account	Account Description	GL Date	Accounting Class	Accounted DR	Accounted CR	Supporting References
Vision Operations (USA)	01-000-1110-0000-000	Operations-No Department-Cash-No Sub Account-No Product	29-Jan-2012	Cash	637,186.73		

Integrating with the EPM Cloud

Oracle Hyperion Financial Data Quality Management, Enterprise Edition can be used as a primary gateway to integrate on-premise and cloud based applications. This feature allows EPM customers to adapt cloud deployments into their existing EPM portfolio. You might use the integration to import data from existing on-premise ERP applications, or synchronize data from on-premise EPM applications. For example, Oracle Hyperion Financial Management customers can add Oracle Hyperion Planning data or a Planning customer can add more Planning applications. In addition, this integration includes writing back from a cloud to an on-premise application or other external reporting applications.

This type of integration supports sources such as:

- On-premise ERP applications from supported source with Oracle Cloud
- On-premise EPM applications with Oracle Cloud
- Oracle Cloud with on-premise EPM applications
- Oracle Cloud with EBS/Peoplesoft GL
- Oracle Cloud with on-premise external sources using custom application
- Oracle General Ledger with on-premise EPM applications
- Oracle Financial Consolidation and Close Cloud applications with on-premise applications

 Note:

Check Rules are not supported for this feature in the 11.1.2.4.200 release.

Watch this tutorial video to learn more about hybrid deployments.



[Tutorial Video](#)

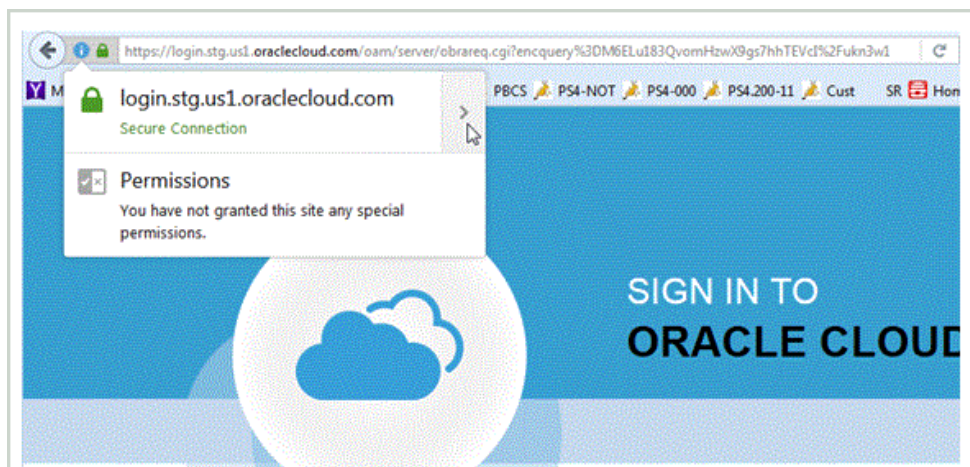
Configuring the Secure Sockets Layer (SSL) Certificate

Use the steps in this procedure to configure the SSL certificate for use with Oracle Enterprise Performance Management Cloud. A SSL is an encryption technology that builds an encrypted connection between the web server and the web browser.

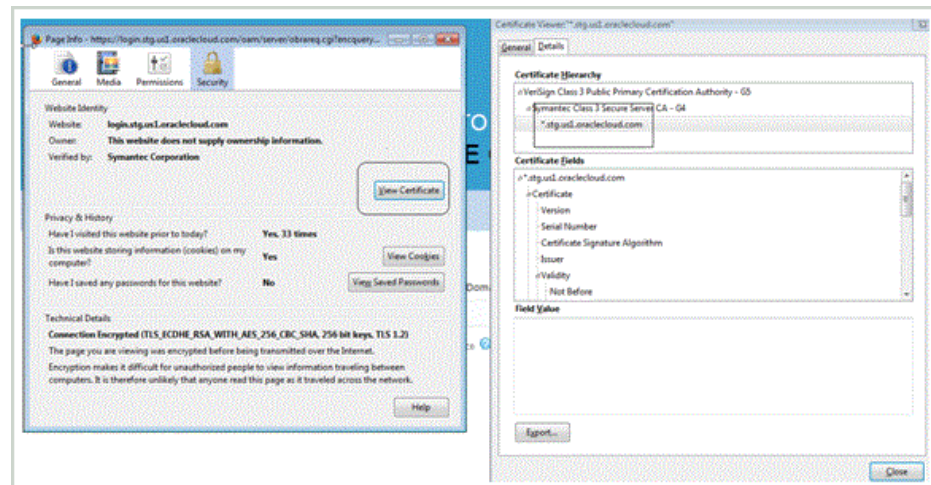
To configure the SSL Certificate:

1. Export the SSL certificate for the EPM Cloud site.

The following example shows download information in a Mozilla Firefox browser.



- a. To download the certificate type in the URL, click the **Lock**.
- b. Select the site and view the certificate.



- c. Click the domain in the hierarchy and save the certificate.
 - d. Assign the name **pbcs.crt**.
2. Copy the certificate file to your server, and import the certificate to the WebLogic keystore.

 Note:

The following instructions are for the Demo Key store in Web Logic. If you are using the production data store, modify parameters accordingly

- a. At a command prompt, type: **cd C:\Oracle\Middleware\wlserver_10.3\server\lib.**
- b. Type: **C:\Oracle\Middleware\jrockit_160_37\jre\bin\keytool -importcert -keystore DemoTrust.jks -storepass DemoTrustKeyStorePassPhrase -file c:/temp/pbcs.crt -alias "pbcs_pod_name"**

```
Administrator: Command Prompt
c:\Oracle\Middleware\wlserver_10.3\server\lib>C:\Oracle\Middleware\jrockit_160_37\jre\bin\keytool -importcert -keystore DemoTrust.jks -storepass DemoTrustKeyStorePassPhrase -file c:/temp/pbcs.crt -alias "pbcs_pod_pbc600buildtest"
Owner: CN=*.stg.us1.oraclecloud.com, OU=FOR TESTING PURPOSES ONLY, O=Oracle Corporation, L=Redwood Shores, ST=California, C=US
Issuer: CN=Symantec Class 3 Secure Server CA - G4, OU=Symantec Trust Network, O=Symantec Corporation, C=US
Serial number: 7a8e8ae7594f51d77994a9c0026c91d5
Valid from: Thu Nov 19 17:00:00 MST 2015 until: Sat Nov 19 16:59:59 MST 2016
Certificate fingerprints:
  MD5: F7:5E:DD:74:09:6A:C1:F6:07:65:2F:3F:BF:91:80:6C
  SHA1: 32:BC:CC:E7:54:2E:BC:E4:A0:30:D0:28:B0:01:EF:2E:7B:F9:55:29
  Signature algorithm name: SHA256withRSA
  Version: 3

Extensions:
#1: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  DigitalSignature
  Key_Encipherment
]
#2: ObjectId: 1.3.6.1.5.5.7.1.1 Criticality=false
AuthorityInfoAccess [
  accessMethod: 1.3.6.1.5.5.7.48.1
  accessLocation: URIName: http://ss.syncd.com,
  accessMethod: 1.3.6.1.5.5.7.48.2
]
```

For the "file," specify the directory and certificate file name.

For the "alias," specify the appropriate name.

3. Using the WebLogic Admin Console, modify two WLS SSL settings.
 - a. Login to the WebLogic Admin Console.
 - b. Select **Lock and Edit**.
 - c. Select **Servers**, and then **ErpIntegrator0**.
 - d. Select **SSL** and then expand **Advanced**.
 - e. Enable **JSEE SSL**.
 - f. In **Hostname Verification**, select **Customer Hostname Verifier**.
 - g. In **Custom Hostname Verifier**, enter **weblogic.security.utils.SSLWLSWildcardHostnameVerifier**.
 - h. Save and activate changes.

The screenshot displays the 'Advanced' tab of the WebLogic Admin Console for the 'ErpIntegrator0' server. The 'SSL' section is expanded, showing various configuration options. Two specific settings are highlighted with red boxes:

- Hostname Verification:** A dropdown menu is set to 'Custom Hostname Verifier'.
- Custom Hostname Verifier:** A text input field contains the value 'weblogic.security.utils.SSLWLSWildcardHostnameVerifier'.
- Use JSSE SSL:** A checkbox is checked.

Other visible settings include 'Export Key Lifespan' (500), 'Use Server Certs' (unchecked), 'Two Way Client Cert Behavior' (Client Certs Not Requested), 'Cert Authenticator' (empty), 'SSLRejection Logging Enabled' (checked), 'Allow Unencrypted Null Cipher' (unchecked), 'Inbound Certificate Validation' (Builtin SSL Validation Only), and 'Outbound Certificate Validation' (Builtin SSL Validation Only). A 'Save' button is located at the bottom left.

4. Restart the FDMEE server.

For each FDMEE server, you need to perform the steps above.

Pre-requisites

The requirements to use a hybrid EPM deployment are:

- Installed and configured Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

- Subscription to the Oracle Cloud
- Details of URL and user name

Setting up the EPM Cloud Deployment

The Oracle Enterprise Performance Management Cloud application is registered with the target application type of the EPM Cloud application, and the deployment mode of **Cloud**. When prompted, specify the connection information. All other setup definitions, such as import format, location, and data rule are defined in the exact same way as any other EPM application.

Loading Data to the EPM Cloud or On-Premise Application

The basic data flow for loading data to the Oracle Enterprise Performance Management Cloud is:

1. Register the EPM Cloud application as a target application in Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

For information on registering an EPM Cloud or on-premise application, see [Registering Target Applications](#).

To do this, on the **Setup** tab, under **Register**, select **Target Application**.

2. Define the import format to map the source to the EPM Cloud or on-premise application.
3. Define the location, data load mapping, and the data load rule in the exact same manner using the normal procedures.

Exporting the EPM Cloud Application

The export data process from Oracle Planning and Budgeting Cloud to Oracle Hyperion Financial Data Quality Management, Enterprise Edition is:

1. Define the import format to map Oracle Enterprise Performance Management Cloud to the FDMEE ERP application for write-back or the FDMEE EPM application (custom or other EPM) for data synchronization.
2. Add a location for the Import Format created above.
3. When you run the data load the on-premise FDMEE generates a DATAEXPORT calc script for EPM Cloud. This calc script is executed in the EPM Cloud and the output file is downloaded to FDMEE.
4. Add maps for the applicable target dimensions.
5. When the file is downloaded, it flows through the write-back/data synchronization workflow process.

Using the Data Load Workbench

The Data Load Workbench feature provides a framework to import, view and verify, and export data from source systems in the Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

Key features of the Workbench include:

- Interactive Load Process with options for Import, Validate, Export, and Check.

- Provision to view Source (All)/Source (Mapped)/Target/Source and Target values
- PTD/YTD Value display for ready reference
- Display options for Valid, Invalid, Ignored, and All Data
- Online and Offline Load process
- Query option for Historical Loads
- Historical Loads Export to Excel
- Drill back to source from the Workbench
- Load, check and post journals for Oracle Hyperion Financial Management applications

 Note:

When you log in with the Run Integration role, these links are visible in the Tasks pane: Data Load Workbench, Data Load, Member Mapping, HR Data Load, Metadata, and Process Detail.

The Data Load Workbench consists of four sections:

- Workflow Grid
- POV Bar—See [Using the POV Bar](#).
- Status
- Data Grid

Workflow Grid

When you select a Workflow step, the following occurs:

Oracle Hyperion Financial Data Quality Management, Enterprise Edition uses fish icons to indicate the status of each step. When a Workflow step is completed successfully, the fish is orange. If the step is unsuccessful, the fish is gray.

 Note:

You can customize the icons that show a "successful process" and a "failed process" by replacing the ProcessSucceeded and ProcessFailed icons in the %EPM_ORACLE_HOME%\epmstatic\aiif\images\general folder.

Processing Data

Step 1: Importing Source Data

The Import from Source feature enables the Oracle Data Integrator to import the data from the source system, performs the necessary transformation, such as import, map

and validate the data. The Import from Source features also enables you to import the source online (immediate processing) or offline (runs in background).

Select this feature only when:

- Running a data load rule for the first time.
- Data in the source system has changed. For example, if you reviewed the data in the staging table after the export, and it was necessary to modify data in the source system.

In many cases, source system data may not change after you import the data from the source the first time. You don't need to keep importing unchanged data.

To import source data:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Workbench**.
2. **Optional:** When you import a source file, Oracle Hyperion Financial Data Quality Management, Enterprise Edition uses the current POV to determine location, category, and period.

To import another source file, you must change the POV. For information on changing the POV, see [Using the POV Bar](#).

3. At the top of the screen, click **Import**.
4. In **Execution Mode**, select the mode of importing the source.
 - online—ODI processes the data in sync mode (immediate processing).
 - offline—ODI processes the data in async mode (runs in background).



Click  to navigate to the Process Detail page to monitor the ODI job progress.

5. Click **OK**.

The Import fish changes to orange.

Step 2: Validating Source Data

Oracle Hyperion Financial Data Quality Management, Enterprise Edition Validation of the source data confirms that all members are mapped to a valid target system account. If there are any unmapped dimension maps within the source file, a validation error occurs. Validation compares the dimension mapping to the source file and identifies unmapped dimensions. The process flow cannot continue until all dimensions are properly mapped.

To run the validation process:

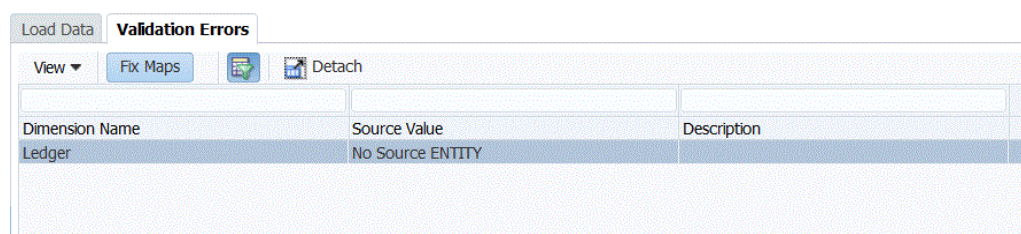
1. On the **Workflow** tab, under **Data Load**, select **Data Load Workbench**.
2. Select **Validate**.

Validation with No Mapping Errors

When validation is successful, the orange Validate fish is displayed in the header of the Oracle Hyperion Financial Data Quality Management, Enterprise Edition screen.

Validation with Mapping Errors

Because newly added dimension members may be unmapped, the addition of dimension members to source systems can produce validation errors. If a dimension member is unmapped, the Validate fish is grey and a Validation Error screen is launched in the Workbench, which shows the number of dimension members that are not mapped (consequently, undefined). You must correct any unmapped dimension members before running the validation again.



Dimension Name	Source Value	Description
Ledger	No Source ENTITY	

To correct conversion-table errors:

1. In the top region of the **Validation** screen, highlight a row that requires correction.
2. Correct any unmapped accounts.

In the top region of the Validation Error screen, the unmapped item is inserted into the Source value, and the bottom region shows all rows with that source value. For example if the top region displays a value of Entity dimension with Source Value 01, then the bottom region should show all rows where ENTITY = '01.'

See [Defining the Import Format Mappings](#).

3. Click **Validate** to refresh the validation form.

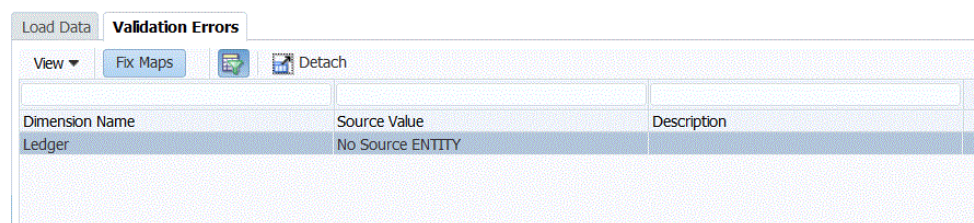
Source data that passes the validation process can be loaded to the target system.

Fixing the Mapping Errors

In the Data Load Workbench, you can view mapping errors and fix them instantly when mapping errors have occurred.

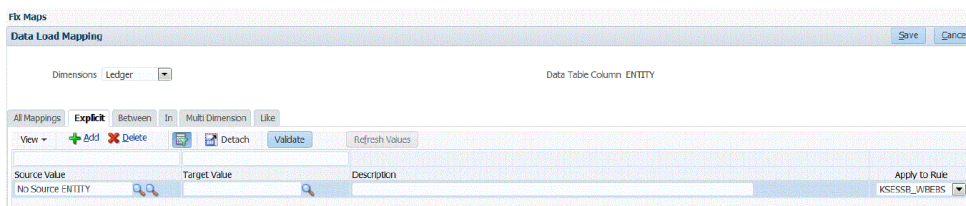
To fix mapping errors:

1. From **Data Load Workbench**, select the **Validation Errors** tab.



Dimension Name	Source Value	Description
Ledger	No Source ENTITY	

2. Select **Fix Maps** to access the mappings.
3. From the **Data Load Mappings** screen, fix any errors.



4. Click **Validate**, and then click **Save**.

Validating Financial Management Data Intersections

When running the validation step, Oracle Hyperion Financial Management you can run the Intersection Check Report to check the data. such as the data intersections (cell status of account, entity, and so on) from the Financial Management target application.

Intersection check reports are generated as part of the data validation step in the data load workflow.

This feature is enabled in the Check Intersection option in Application Options.

To run an intersection check:

1. In the top region of the **Validation** screen, click **Generate Intersection Check Report**.
2. When prompted, save or open the Intersection Check Report.
3. Correct validation errors and rerun the validation step.

See [Intersection Check Report Errors](#).

Step 3: Exporting Data to Target Applications

After the source data has passed the validation process, use the Export option to export data to a target application. Select this option after you have reviewed the data in the data grid and are sure you want to export it to the target application.

When exporting data for Oracle Hyperion Planning and Oracle Essbase, you can store, add, and subtract data. For Planning and Essbase, you can override all data.

For Oracle Hyperion Financial Management, you can merge, accumulate, replace, and replace by security data.

When you use Oracle Hyperion Enterprise Performance Management System Lifecycle Management to export mapping rules, any related mapping scripts are included.

The export of mapping rules to a CSV or Excel format does not include any scripting.

To submit the data load rule:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Workbench**.
2. **Optional:** When you import a source file, Oracle Hyperion Financial Data Quality Management, Enterprise Edition uses the current POV to determine location, category, and period and conducts the following process: To import another source file, you must change the POV. See [Using the POV Bar](#).
3. At the top of the screen, click **Export**.

- drill through to the source data
- view mapping details

You perform tasks on the data grid by selecting options on the Table Action including:

- [Viewing Data](#)
- [Formatting Data](#)
- [Showing Data](#)
- [Opening Loaded Data in Microsoft Excel](#)
- [Querying by Example](#)
- [Freezing Data](#)
- [Detaching Data](#)
- [Wrapping Text](#)
- [Attaching Cell Text and Documents to a Data Cell](#)

Viewing Data

The View data provides multiple ways to view data including:

Table—Selects the source or target data to display in the grid:

- Source (All)—Shows both mapped and unmapped source dimensions (ENTITY, ACCOUNT, UD1, UD2,... AMOUNT).
- Source (Mapped)—Shows only mapped source dimensions.
- Target—Shows only target dimensions (ENTITYX, ACCOUNTX, UD1X, UD2X, ...AMOUNTX).
- Source and Target—Shows both source and target dimensions (ENTITY, ENTITYX, ACCOUNT, ACCOUNTX, UD1, UD1X, AMOUNT, AMOUNTX).

Columns—Selects the columns to display in the data:

- Show All
- Entity
- Account
- Version
- Product
- Department
- STAT
- Amount
- Source Amount



Note:

For Oracle E-Business Suite and PeopleSoft, the Account Descriptions is also available for viewing.

Freeze/Unfreeze—Locks a column in place and keeps it visible when you scroll the data grid. The column heading must be selected to use the freeze option. To unfreeze a column, select the column and from the shortcut menu, select *Unfreeze*.

Detach/Attach—Detaches columns from the data grid. Detached columns display in their own window. To return to the default view, select *View*, and then click *Attach* or click *Close*.

Sort—Use to change the sort order of columns in ascending or descending order. A multiple level sort (up to three levels and in ascending and descending order) is available by selecting *Sort*, and then *Advanced*. From the *Advanced Sort* screen, select the primary "sort by" column, and then the secondary "then by" column, and then the third "then by" column.

The search fields that are displayed in the advanced search options differ depending on what artifact you are selecting.

Reorder Columns—Use to change the order of the columns. When you select this option, the *Reorder Columns* screen is displayed. You can select a column, and then use the scroll buttons on the right to change the column order.

Query by Example—Use to toggle the filter row. You can use the filter row to enter text to filter the rows that are displayed for a specific column. You can enter text to filter on, if available, for a specific column, and then press [Enter]. To clear a filter, remove the text to filter by in the text box, then press [Enter]. All text you enter is case sensitive.

Formatting Data

You can resize the width of a column by the number pixel characters or a percentage. You can also wrap text for each cell automatically when text exceeds the column width.

To resize the width of a column:

1. Select the column to resize.
2. From the table action bar, select **Format**, and then **Resize**.
3. In the first **Width** field, enter the value by which to resize.

You can select a column width from 1 to 1000.

4. In the second **Width** field, select **pixel** or **percentage** as the measure to resize by.
5. Select **OK**.

To wrap the text of a column:

1. Select the column with the text to wrap.
2. From the table action bar, select **Format**, and then **Wrap**.

Showing Data

You can select the type of data to display in the data grid including:

- **Valid Data**—Data that was mapped properly and is exported to the target application.
- **Invalid Data**—One or more dimensions that was not mapped correctly and as a result, the data is not exported to target.

- Ignored Data—User defined explicit map to ignore a source value when exporting to target. This type of map is defined in the member mapping by assigning a special target member with the value of **ignore**.
- All Data—Shows all valid, invalid and ignored data.

To show a type of data:

1. Select **Show**.
2. Select from one of the following:
 - Valid Data
 - Invalid Data
 - Ignored Data
 - All Data

Drilling Through to Source Data and Viewing Mappings

When data has been displayed in the Data Load Workbench, you can drill through to the source, view mappings and open the source document.



Note:

If the source system is Oracle E-Business Suite/PeopleSoft and you have metadata rules, then the drill region is created based on the metadata rule. Otherwise, it is created based on the target members in the data load mappings. For Year, Period, and Scenario, Oracle Hyperion Financial Data Quality Management, Enterprise Edition uses audit information to create the drill region.

To drill through to the source mapping:

1. Select the type of data to display in the data grid.
See [Showing Data](#).
2. In **Source Amount** column, select an amount.
3. Click the source amount link and select **Drill through to source**.

To view the source mapping detail:

1. Select the type of data to display in the data grid.
See [Showing Data](#).
2. From the **Source Amount** column, select an amount.
3. Click the source amount link and select **View Mappings**.

Mapping Details					
Dimension Name	Type	Source	Target	Rule Name	Description
Account	LIKE	*	MKA*	DEFAULT	System Generated M
Entity	LIKE	*	MKE*	DEFAULT	System Generated M
Version	LIKE	*	BU Version_1	Default	

Opening Loaded Data in Microsoft Excel

When reviewing data in the workbench, users can drill down from the amount to the Enterprise Resource Planning (ERP) source system. In the source system the data is displayed in the granularity with which it was loaded.

You can open loaded data in Microsoft Excel and review how the data is defined.

To open loaded data in Microsoft Excel:

1. From the table action bar, click .
2. Open the loaded data in Microsoft Excel.

Querying by Example

Use the Query by Example feature to filter rows that are displayed for a specific column. You can enter text to filter on, if available, for a specific column, and then press [Enter]. To clear a filter, remove the text to filter by in the text box, then press [Enter]. All text you enter is case sensitive.

To query by example:

1. From the table action bar, click  to enable the filter row.
The filter row must appear above the columns to use this feature.
2. Enter the text by which to filter the values in the column and press [Enter].

Note:

When entering text to filter, the text or partial text you enter is case-sensitive. The case must match exactly. For example, to find all target applications prefixed with "HR," you cannot enter "Hr" or "hr."

Freezing Data

Use the Freeze feature to lock a column in place and keeps it visible when you scroll the data grid.

To freeze a column:

1. Select the column to freeze.
2. From the table action bar, click .

To unfreeze a column:

1. Select the frozen column.
2. On the shortcut menu, select **Unfreeze**.

Detaching Data

Use the Detach feature to detach columns from the data grid. When you detach the grid, columns display in their own window. To return to the default view, select **View**, and then click **Attach** or click **Close**.

To detach columns:

1. Select the column to detach.
2. From the table action bar, click .

The data grid is displayed in a separate window.

To reattach columns to the data grid:

1. Select the column to reattach.
2. From the table action bar, select **View**, and then **Attach**.

Wrapping Text

You can wrap text for each cell automatically when text exceeds the column width.

To wrap text for a column:

1. Select the column with the text to wrap.
2. Click .

Attaching Cell Text and Documents to a Data Cell

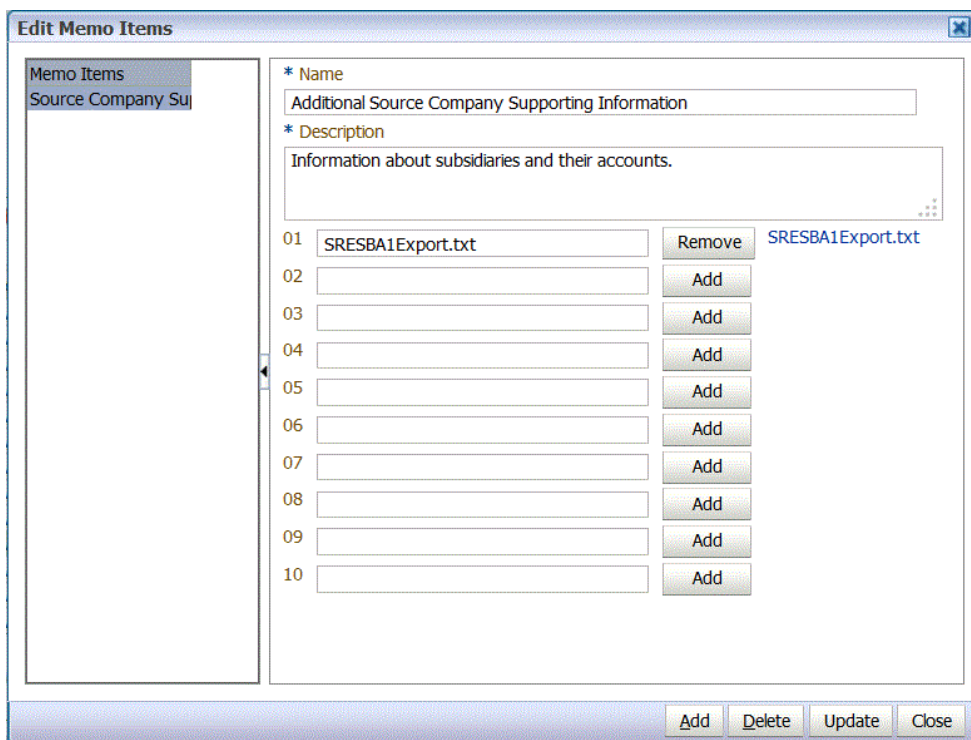
The cell text feature enables you to attach text and documents to a data cell. Multiple instances of cell text can be added as needed. Oracle Hyperion Financial Data Quality Management, Enterprise Edition archives documents in EPM_ORACLE_HOME/products/FinancialDataQuality/data directory. Cell text can only be exported in Oracle Hyperion Financial Management applications.

 Note:

FDMEE does not load multiple cell text to an intersection in Financial Management. If a load using an append mode is run and new cell text is added to an intersection that already has cell text, the old cell text is replaced by the new cell text and not appended.

To assign cell text and attach documents:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Workbench**.
2. In **Data Load Workbench**, select the data cell.
3. From the memo column () in the column heading, click the memo link () to the left the data cell.
4. From **Edit Memo Items**, click **Add**.
5. In the **Name** field, enter a name of the memo.
6. In the **Description** field, enter a description of the memo.
7. Click **Add** (to the right of an Attachment field).
8. On the **Select** screen, browse and select an attachment, and then click **OK**.



9. Click **Update**.
10. Click **Close**.

11. **Optional:** To remove an attachment, click **Remove** (to the right of an Attachment field).

Intersection Check Report Errors

When Intersection Check reports are generated as part of the data validation step, if errors are encountered, the Intersection check report provides information on the errors.

There are four Intersection Check Report errors:

- Invalid Intersection
- Cannot Write
- Locked Intersection
- Invalid Member

Invalid Intersection (highlighted in red)

Invalid intersection errors occur due to the following errors:

- The intersection is not valid.
- The ICP member is not valid for the account. The ICP must be a child of "TopCustom" in the ICP hierarchy.
- The account does not allow intercompany activity. The ICP member value must be set to "[ICP None]" or the account must be changed.
- The entity does not allow intercompany activity. The ICP member value must be set to "[ICP None]" or a valid ICP entity.
- The custom member is not valid for the account. The custom member must be a child of "TopCustom" in the custom hierarchy.

Cannot Write (highlighted in purple)

Cannot write errors occur due to the following:

- No read or write access. (Oracle Hyperion Financial Management cell status is no read access or no write access.)
- Financial Management cell status can write and support line items, but the cell does not support IC transactions.
- Financial Management cell status is derived.
- Financial Management cell status is parent level input.

Locked Intersection (highlighted in green)

Locked intersection. (Financial Management cell status is locked.)

Invalid Member (highlighted in orange)

Invalid member mapping error due to the following:

- The target member is not valid in Financial Management or the user does not have access to it in Financial Management.
- Intercompany with self is restricted. The ICP dimension value must not equal the entity dimension value.

Viewing Process Details

You use the Process Details page to view submitted rule status and logs.

To view data rule process details:

1. On the **Workflow** tab, under **Monitor**, select **Process Details**.

The Process Details page is displayed, showing processes for all source systems. The following columns are displayed for each process:

- **Process ID**—An automatically generated identification number
- **Status**—Displays a visual indicator for the status of the process. You can rest the cursor over the icon to view a Screen Tip. Available statuses:
 -  —Rule Processed Successfully
 -  —Rule Execution did not complete successfully
- **Log**—Click **Show** to display the log file.
- **Location**—Displays the location name
- **Process Name**—Type of process

Types of processes include:

- **Data Load**—Initiated when you run a data load rule.
- **Metadata Load**—Initiated when you run a metadata load rule.
- **HR Load**—Initiated when you run an HR data load rule.
- **Purge Process**—Initiated when you remove an artifact, such as a target application or source system.
- **Initialize Source System**—Initiated when you initialize a source system.
- **Rule Name**—Name of the rule
- **Source System**—Name of the source system
- **Accounting Entity**—Name of the source accounting entity
- **Target Application**—Name of the target application
- **ODI Session Number**—The session number in Oracle Data Integrator. You can use this to look up a session in Oracle Data Integrator.

 Note:

The ODI Session number is present in Process Details only when the data is processed during an offline execution.

- **Job ID**—The Oracle Hyperion EPM Architect job ID
- **Process By**—The user ID who initiated the process
- **Reset Status**—Resets the status to failed if a process continues to stay in a running status for a long period of time.

- **Link**—Shows the log information for the process step. In the case of File Import, it shows skipped rows, and in the case of export to Oracle Hyperion Planning, it shows rejected rows and so on.
2. Select a process to display the details:
 - **Status**—For each process step, the status is displayed. You can troubleshoot a problem by viewing at which point the process failed.
 - **Process Step**—Displays the steps in the process.
 - **Process Start Time**—Time that the process step started.
 - **Process End Time**—Time the process step ended.
 - **Log**—If a log is available, you can click **Show** to display the log contents.
 3. **Optional:** To filter the rows that are displayed, ensure that the filter row appears above the column headers. (Click  to toggle the filter row.) Then, enter the text to filter.

You can filter by:

- Process ID
- Location
- Rule Name
- Source System
- Accounting Entity
- Target Application

 Note:

When entering text to filter, the text or partial text that you enter is case sensitive. For example, to find all target applications prefixed with “HR,” you cannot enter “Hr” or “hr.” For additional information on filtering, see [FDMEE User Interface Elements](#).

Integrating with the Oracle Financials Cloud

You can integrate Oracle General Ledger data from the Oracle Financials Cloud with your EPM application if you use Oracle Financials Cloud Release 11 or higher. This integration enables you to simply pick the desired source ledger from the Oracle Financials Cloud, set up a few simple mappings and then push a button to pull the data into the EPM application. This integration can be run manually, or scheduled for a specific time.

 Note:

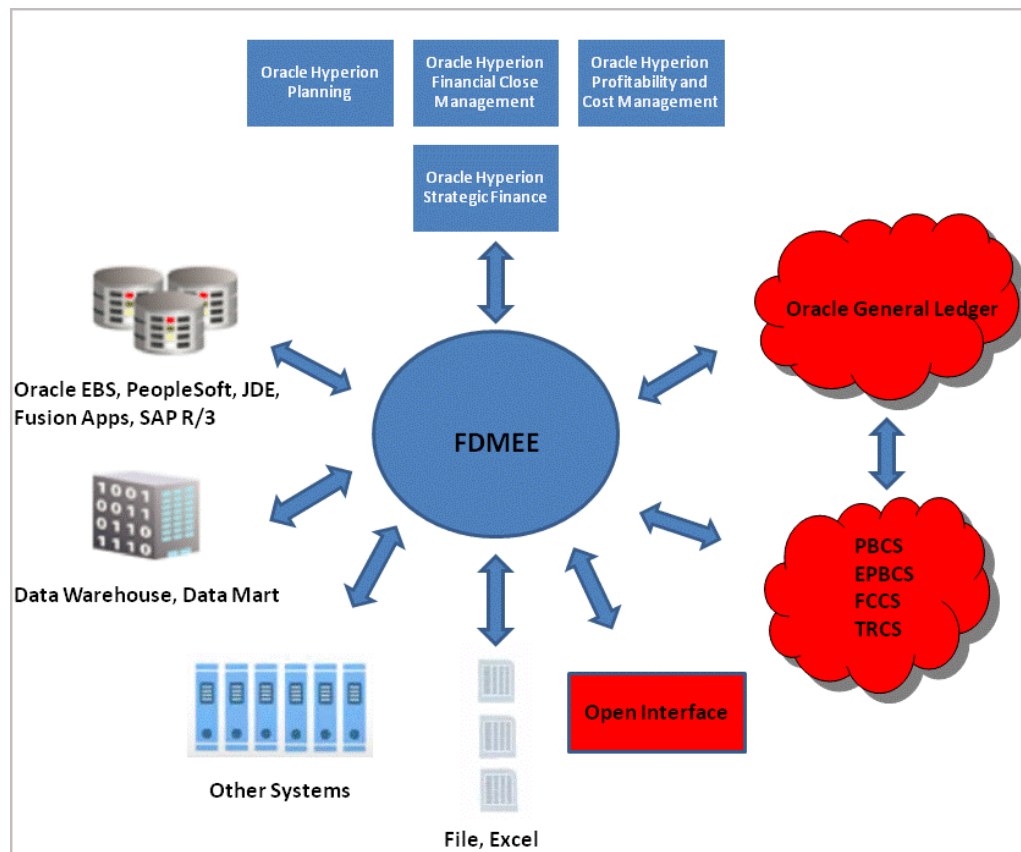
The Average Daily Balances (ADB) ledger is not supported in the current integration.

Note:

Oracle Hyperion Financial Data Quality Management, Enterprise Edition also supports the Financials Accounting Hub (FAH) and the Financial Accounting Hub Reporting Cloud Service (FAHRCS) as part of its integration with the Oracle General Ledger.

The integration sets up the drill definition automatically.

FDME facilitates not only data loading, but write-back to the Oracle Financials Cloud.



Integration Process Description

At a high level, this is how you integrate Oracle General Ledger data from the Oracle Financials Cloud with your EPM application:

1. Set up the Oracle General Ledger source system from the Oracle Financials Cloud connection information and initialize the source system.

The initialize process brings over the Oracle General Ledger data into the EPM application as Oracle Essbase cubes. Each Essbase target application represents a chart of accounts definition from the source Oracle General Ledger.

See [Configuring a Source Connection](#).

2. Create the EPM target application that requires the Oracle General Ledger data from the Oracle General Ledger source system.

3. Create the target application that requires data from one or more source systems.

If you are loading data from an Oracle General Ledger application to an EPM application, add the EPM application as the target application type (for example, add Oracle Hyperion Planning as the target application type.)

4. Set up the integration mapping between Oracle General Ledger and the EPM application dimensions in Oracle Hyperion Financial Data Quality Management, Enterprise Edition by building an import format.

See [Working with Import Formats](#) in this section

5. Define the location used to associate the import format with the Oracle General Ledger segments.

See [Defining Locations](#) in this section

6. Create category mapping for scenario dimension members in the EPM application to which Oracle General Ledger balances are loaded.

See [Defining Category Mappings](#) in this section.

7. Define data load mapping to convert the chart of accounts values from the Oracle General Ledger to dimension members during the transfer.

See [Data Load Mapping](#) in this section.

8. Define a data rule with the necessary filters and execute the rule.

A default filter is provided that includes all dimensions of the Essbase cube. The cube may have duplicate members so fully qualified member names are required. The Essbase cubes work off the Oracle General Ledger segments, and there is a one to many relationships of Chart of Accounts to ledgers in the Oracle General Ledger.

FDME creates filters when a rule is created. You can modify the filters as needed but cannot delete them. (If the filters are deleted, FDME recreates the default values). For information about these filters, see [Adding Filters for Data Load Rules](#).

The process extracts and loads the data from Oracle Financials Cloud to FDME.

See [Adding Data Load Rules](#)

9. **Optional:** Write back the data to the Oracle Financials Cloud.

To write back data to Oracle Financials Cloud from a Planning or an Oracle Enterprise Planning and Budgeting Cloud source system, set up a data rule. In this case, the filters are applied against the Planning or Oracle Enterprise Planning and Budgeting Cloud application.

Optionally, you can write back budget data from a Planning to a flat file using a custom target application. This output file may be used to load data to any other application.

Configuring a Source Connection

To begin integrating the Oracle General Ledger with the Oracle Enterprise Performance Management Cloud, you first create and register the source system with the type "Oracle Financial Cloud."

After the source system and connection information are specified, you initialize the source system to copy the Oracle General Ledger Chart of Account information to Oracle Hyperion Financial Data Quality Management, Enterprise Edition as one of

many Oracle Essbase target applications. The download is an integration with the Essbase database of the Oracle Hyperion Planning application. You can see the actual cubes in Oracle Smart View for Office.

To begin integrating the Oracle General Ledger with an EPM application, you first create and register the source system with the type “Oracle Financial Cloud.”

To add a source system:

1. On the **Setup** tab, under **Register**, select **Source System**.
2. In **Source System**, click **Add**.
3. Enter the source system details:
 - a. In **Source System Name**, enter the source system name.
Enter the Oracle General Ledger name to use for the file, such as “General Ledger” or “Oracle General Ledger Financials.” If you are also using this source for write-back, make sure the source system name does not include any spaces.
 - b. In **Source System Description**, enter a description of the source system.
 - c. In **Source System Type**, select **Oracle Financials Cloud**.
 - d. In **Drill Through URL**, specify one of the following Oracle Financials Cloud release URL formats:
 - **R13**—system uses the Oracle Financials Cloud Release 13 URL format.
 - **R12**—system uses the Oracle Financials Cloud Release 12 and earlier URL format.
 - (Null)—system uses the Oracle Financials Cloud Release 12 and earlier URL format.

If you need to overwrite the server in addition to specify the release URL format, specify one of the following Oracle Financials Cloud release URL formats:

- **R13@https://server**—system uses the Oracle Financials Cloud Release 13 URL format and your server.
 - **R12@https://server**—system uses the Oracle Financials Cloud Release 12 and earlier URL format and your server.
- e. Leave the **Fusion Budgetary Control** field unchecked.

The screenshot shows a 'Details' form with the following fields and values:

- * Source System Name:** Oracle General Ledger
- * Source System Type:** Oracle Financials Cloud (selected from a dropdown)
- Source System Description:** R13
- Drill Through URL:** (empty text area)
- Fusion Budgetary Control:** ☐ (unchecked)

- f. In **Application Filter**, specify any filter(s) condition to limit the number of Essbase applications returned when you initialize the source system.
You can specify a single filter condition or multiple filters conditions. If you use multiple filter condition, separate each filter condition by a comma (.).
When you specify an application name as a filter condition, you can specify the full Oracle Financials Cloud application name, wild card, or a wild card for a single character as shown below.

- Vision (Full Name)
- VF* (Wild Card)
- VF??COA (Wild Card for single char_

Oracle General Ledger : Application Filter

Application Filter

When you run the initialize process, the system imports all the applications that match the filter condition. If no filters are provided, all applications are imported.

4. Click **Configure Source Connection**.

The Configure Source Connection screen is used to configure the connection to the Oracle Financials Cloud.

The source connection configuration is used to store the Oracle Financials Cloud user name and password. It also stores the WSDL connection for the Oracle Financials Cloud user name and password.

Configure Source Connection

User Name

Password

Fusion Web Services URL

5. In **User Name**, enter the Oracle Financials Cloud user name.

Enter the name of the Oracle Financials Cloud user who launches the process requests to send information between Planning and the Oracle Financials Cloud. This user must have an assigned Oracle General Ledger job role such as "Financial Analyst," "General Accountant," or "General Accounting Manager."

6. In **Password**, enter the Oracle Financials Cloud password.

You must update this password anytime you change your Oracle Financials Cloud password.

7. In **Fusion Web Services URL**, enter the URL for the Fusion web service.

For example, you might specify: `https://server/publicFinancialCommonErpIntegration/ErpIntegrationService?WSDL`

Replace "fs" with "fin" in the URL from the one that is used to log on.

8. Click **Test Connection**.

9. Click **Configure**.

The confirmation "Source system [source system name] configuration has been updated successfully" is displayed.

10. On the **Source System** screen, click **Initialize**.

Initializing the source system fetches all metadata needed in FDMEE, such as ledgers, chart of accounts, and so on. It is also necessary to initialize the source system when there are new additions, such as chart of accounts, segments/chartfields, ledgers, and responsibilities in the source system.

The initialize process may take a while, and you can watch the progress in the job console.



Note:

When re-initializing an Oracle General Ledger source, application period mappings are reset/removed from the system. If specific period mappings are required, then use the source period mapping tab to specify the period mappings.

11. Click **Save**.

After you add a source system, select the source system in the table, and the details are displayed in the lower pane.

The initialize process may take a while, so the user can watch the progress in the job console.

Working with Import Formats

The import format enables you to set up the integration mapping between the Oracle General Ledger (Chart of Accounts) segments and the dimensions in the EPM application.

When you select the source and target, Oracle Hyperion Financial Data Quality Management, Enterprise Edition populates the source and target columns automatically.



Note:

Oracle General Ledger creates one Essbase cube per Chart of Account/Calendar combination. In this case, you can use the same import format to import data from Ledgers sharing this Chart of Accounts. Ledgers can be specified as a filter in the data load rule.

You work with import formats on the Import Format screen, which consists of three sections:

- **Import Format Summary**—Displays common information relevant to the source and target applications.
- **Import Format Detail**—Enables you to add and maintain import format information.
- **Import Format Mappings**—Enables you to add and maintain import format mapping information.

To add an import format for an Oracle General Ledger based source system:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.

2. In the **Import Format** summary task bar, select **Add**.
In the upper grid of the Import Formats screen, a row is added.
3. In **Name**, enter a user-defined identifier for the import format.
You cannot modify the value in this field after a mapping has been created for this import format.
4. In **Description**, enter a description of the import format.
5. In **Source**, select the Oracle General Ledger Chart of Accounts from the drop down.
6. In **Target**, select the EPM target application.
7. **Optional:** In **Expression**, add any import expressions.
FDMEE provides a set of powerful import expressions that enable it to read and parse virtually any trial balance file into the FDMEE database. Advanced expressions are entered into the Expression column of the field grid of the Import Formats screen. Import expressions operate on the value that is read from the import file.
For more information, see [Adding Import Expressions](#).
8. Click **Save**.

Defining Locations

A location is the level at which a data load is executed in Oracle Hyperion Financial Data Quality Management, Enterprise Edition. Each location is assigned an import format. Data load mapping and data load rules are defined per location. You define locations to specify where to load the data. Additionally, locations enable you to use the same import format for more than one target application where the dimensionality of the target applications is the same. However; if you are using multiple import formats, you must define multiple locations.



Note:

You can create duplicate locations with the same source system and application combination.

To create, edit, and delete import locations:

1. On the **Setup** tab, under **Integration Setup**, select **Location**.
2. In **Location**, click **Add**.
3. From **Location Details**, in **Name**, enter the location name.
4. From **Import Format**, enter the import format.

The import format describes the source system structure, and it is executed during the source system import step. A corresponding import format must exist before it can be used with a location.

Additionally:

- Source name is populated automatically based on the import format.

- Target name is populated automatically based on the import format.

You can also click  and select an import format.

5. In **Functional Currency**, specify the currency of the location.

**Note:**

You must specify the budget currency of the control budget to which budget is written back.

6. In **Parent Location**, enter the parent assigned to the location.

Parent mapping are used to share mappings with other locations. Enter mappings at the parent location, and the related locations can use the same mappings. Multiple locations can share a parent. This feature is useful when multiple locations use one chart of accounts. Changes to a child or parent mapping table apply to all child and parent locations.

**Note:**

If a location has a parent, the mappings are carried over to the child. However; changes to mapping can only be performed on the parent location.

7. **Optional:** In **Logic Account Group**, specify the logic account group to assign to the location.

A logic group contains one or more logic accounts that are generated after a source file is loaded. Logic accounts are calculated accounts that are derived from the source data.

The list of values for a logic group is automatically filtered based on the Target Application under which it was created.

8. **Optional:** In **Check Entity Group**, specify the check entity group to assign to the location.

When a check entities group is assigned to the location, the check report runs for all entities that are defined in the group. If no check entities group is assigned to the location, the check report runs for each entity that was loaded to the target system. FDMEE check reports retrieve values directly from the target system, FDMEE source data, or FDMEE converted data.

The list of values for a check entity group is automatically filtered based on the target application under which it was created.

9. **Optional:** In **Check Rule Group**, specify the check rule group to assign to the location.

System administrators use check rules to enforce data integrity. A set of check rules is created within a check rule group, and the check rule group is assigned to a location. Then, after data is loaded to the target system, a check report is generated.

The list of values for a check rule group is automatically filtered based on the target application under which it was created.

10. Click **Save**.

11. **Optional:** Perform these tasks:

- To edit an existing location, select the location to modify, and then make changes as necessary. Then, click **Save**.
- To delete a location, click **Delete**.

When a location is deleted, the location is removed from all other FDMEE screens, such as Data Load.

Tip:

To filter by the location name, ensure that the filter row is displayed above the column headers. (Click  to toggle the filter row.) Then, enter the text to filter.

You can filter locations by target application using the drop down at the top of the screen.

Defining Category Mappings

You define category mappings for categorizing and mapping source system data to a target EPM Scenario dimension member. For example, you may have a Scenario dimension member called Actuals for storing actual balances from an Oracle General Ledger application. In an Oracle Hyperion Planning application, the same source system data is stored using the Scenario dimension member "Current." In Oracle Hyperion Financial Data Quality Management, Enterprise Edition, you can create one category mapping to give both one name to represent their respective scenarios.

You create category mapping for the scenario dimension member in Planning from which budget is written back.

1. On the **Setup** tab, under **Integration Setup**, select **Category Mapping**.
2. Select **Global Mapping**.
3. Click **Add**.
A blank entry row is displayed.
4. In **Category**, enter a name that corresponds to the Planning application Scenario dimension member from which you want to load data.
5. In **Target Category**, enter the name of the Planning Scenario dimension members from which you want to load data.
6. Click **Save**.

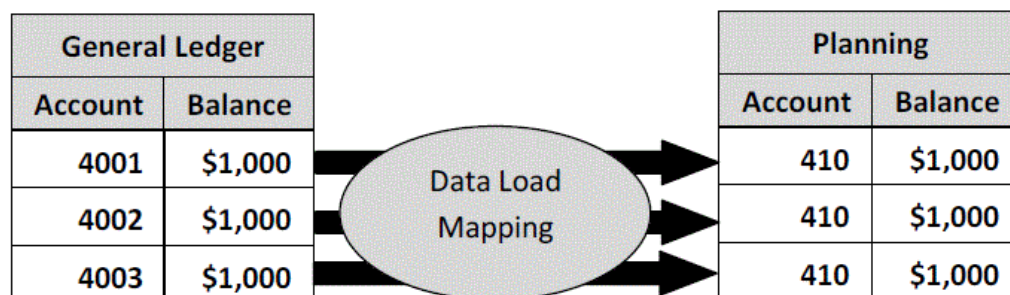
Data Load Mapping

Data load mappings convert the chart of accounts values from the Oracle General Ledger to the dimension members of the EPM application during the transfer. This allows Oracle Hyperion Financial Data Quality Management, Enterprise Edition to categorize Oracle General Ledger balances.

In the following example, based on the chart of accounts segment, the Oracle General Ledger breaks down administrative expenses as a range of accounts from 4001-4003.

In Oracle Hyperion Planning, budgeting for administrative expenses is all done for dimension value 410, Administrative Expenses.

The data load mapping maps Oracle General Ledger accounts in the range 4001-4003 actual amounts to 410 administrative expenses in Planning.



Differences in Planning dimension values and Oracle General Ledger Chart of Accounts values may occur. Additionally segments of the Chart of Accounts may not be used when budgeting.

For example, the data load mapping can be used to use zero values (such as "0000" for sub-account) for the chart of accounts segments that are not used for budgeting.

To define data load mappings:

1. On the **Workflow tab**, under **Data Load**, select **Data Load Mapping**.

2. From the **Dimensions** drop-down, select the dimension that you want to map. The "*" represents all values. Data load mappings should be based upon your EPM application requirements.

When there is no update to the Oracle General Ledger value prior to the load, it is still necessary to create the data load mapping for the dimensions to instruct FDMEE to create the target values.

At a minimum, map values for the "Account" and "Entity" dimensions since those are transferred from Oracle General Ledger.

If you are transferring additional chart segments you must provide a mapping for each destination dimension.

3. To map all General Ledger accounts to the EPM application "as is" without any modification, in **Source Value**, enter *, and from **Target Value**, enter *.
4. Select the **Like** tab.
5. In **Source Value**, enter * to indicate that all values should use the mapping. These are the values from the Oracle General Ledger Chart of Accounts. Enter the values directly.

6. In **Target Value**, enter the value for the accounting scenario to use to load the budget information.
Enter the values that should be used in the EPM application to store the Oracle General Ledger actual balances that are transferred.
7. In **Rule Name**, enter the name of the data load rule used to transfer budget amounts to the Oracle General Ledger.

 Note:

Rules are evaluated in rule name order, alphabetically. Explicit rules have no rule name. The hierarchy of evaluation is from Explicit to (In/Between/Multi) to Like.

8. In **Description**, enter a description of the mapping.
For example, enter a description such as “Map to the General Ledger.”
9. **Optional:** In **Apply to Rule** to apply the mapping only to the specific data rule in the location.
See [Creating Member Mappings](#).

Adding Data Load Rules

After you define member mappings for the location, define data load rules for ledgers or business units in your source system to extract the data from the Oracle General Ledger and move it to the EPM application.

Data load rules are defined for locations that you have already set up. Data load rules are specific to locations. You can create multiple data load rules for a target application so that you can import data from multiple sources into a target application.

The data load rule is created once, but used each time there is a transfer.

To create a data load rule for synchronization:

1. On the **Workflow tab**, under **Data Load**, select **Data Load Rule**.
2. From the **POV Bar**, select the location to use for the data load rule.
Data load rules are processed within the context of a point of view. The default point of view is selected automatically. The information for the point of view is shown in the POV bar at the bottom of the screen.
3. Click **Add**
4. In **Name**, enter the name of the data load rule.
5. In **Category**, leave the default category value.
The categories listed are those that you created in the Oracle Hyperion Financial Data Quality Management, Enterprise Edition setup. See [Defining Category Mappings](#).
6. In **Period Mapping Type**, select the period mapping type for each data rule.
Valid options:
 - **Default**—The Data Rule uses the Period Key and Prior Period Key defined in FDMEE to determine the source General Ledger periods mapped to each FDMEE period included in a Data Rule execution.

- **Explicit**—The Data Rule uses the Explicit period mappings defined in FDMEE to determine the source General Ledger periods mapped to each FDMEE period included in a data load rule execution. Explicit period mappings enable support of additional Oracle General Ledger data sources where periods are not defined by start and end dates.
- Click **Save**.

Processing Oracle General Ledger Adjustment Periods

You can include adjustment periods from an Oracle General Ledger source system in the Oracle Financials Cloud when loading balances to an EPM application.

Adjustment periods are additional periods that are related to regular periods from the source. An "adjustment period" refers to any accounting period set up to adjust balances prior to the closing period of the year. These periods are adjusted to per12 and consequently are referred to as "per13." Typically dates within the adjustment period overlap regular accounting periods. A customer might use a "Year Open Period" that refers to the first period in the accounting calendar to adjust last year's balance carried forward amount. Additionally, the customer can set up the last period of the accounting calendar as the "Year Close Period" to adjust transactions made in the current accounting calendar.

In Oracle Hyperion Financial Data Quality Management, Enterprise Edition, adjustments are processed in Period Mappings where you indicate how the adjustment period maps to the period in the target application. The method for how adjustments get processed is specified in the data load rule. This feature enables you to map Oracle General Ledger source periods in FDMEE simply by pointing to the calendar and periods from the Oracle General Ledger application to the period in the EPM application.

When setting up the data load rule, you can either load to regular and adjustment periods when an adjustment period mapping exists, or load an adjustment period only when an adjustment period mappings exists.

For example, when you map period 13 to December/Period 12, and select the Include Adjustment Period option, then the following occurs:

- For YTD balances, period 13 becomes the ending balance.
- For PTD balances, period 13 and December/Period12, are added.

To include adjustment periods from an Oracle General Ledger source system:

1. On the **Setup** tab, under **Integration Setup**, select **Period Mapping**.
2. Select the **Source Mapping** tab.
3. From **Source System**, select the Oracle General Ledger source system.
4. From **Target Application**, select the EPM application to which the adjustment applies.
5. From **Mapping Type**, select **Adjustment**.
6. Click **Add**.
7. In **Source Period Key**, specify the last day of the month to be mapped from the Oracle General Ledger source system.

Use the date format based on the locale settings for your locale. For example in the United States, enter the date using the **MM/DD/YY** format.

You can also click  and browse to and select the source period key.

When you select the **Source Period Key**, FDMEE populates the **Source Period** and **Source Period Year** fields automatically.

8. In **Adjustment period**, specify the name of the adjustment period from the Oracle General Ledger source.

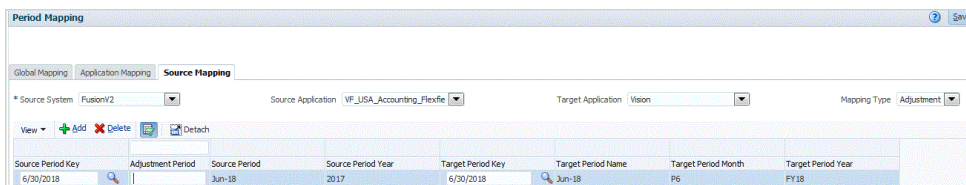
For example, if the adjustment period from the Oracle General Ledger is Adj-Dec-16, then enter **Adj-Dec-16** in this field.

9. In **Target Period Key**, specify the last day of the month to be mapped from the target system.

Use the date format based on the locale settings for your locale. For example in the United States, enter the date using the **MM/DD/YY** format.

You can also click  and browse to and select the target period key.

When you select the **Target Period Key**, FDMEE populates the **Target Period Name**, **Target Period Month**, and **Target Period Year** fields automatically.



Source Period Key	Adjustment Period	Source Period	Source Period Year	Target Period Key	Target Period Name	Target Period Month	Target Period Year
6/30/2018	Jun-18	Jun-18	2017	6/30/2018	Jun-18	P6	FY18

10. Click **Save**.
11. On the **Workflow tab**, under **Data Load**, select **Data Load Rule**.
12. From the **POV Bar**, select the location to use for the data load rule.

Data load rules are processed within the context of a point of view. The default point of view is selected automatically. The information for the point of view is shown in the POV bar at the bottom of the screen.

13. Click **Add**.
14. In **Name**, enter the name of the data load rule.
15. In **Category**, specify the default category value.

The categories listed are those that you created in the FDMEE setup.

See [Defining Category Mappings](#).

16. In **Period Mapping Type**, select the period mapping type for each data rule.

Valid options:

- **Default**—The Data Rule uses the Period Key and Prior Period Key defined in FDMEE to determine the source General Ledger periods mapped to each FDMEE period included in a Data Rule execution.
- **Explicit**—The Data Rule uses the Explicit period mappings defined in FDMEE to determine the source General Ledger periods mapped to each FDMEE period included in a data load rule execution. Explicit period mappings enable support of additional Oracle General Ledger data sources where periods are not defined by start and end dates.

17. From **Include Adjustment Period**, select one of the following options for processing adjustment periods:

- No—Adjustment periods are not processed. The system processes only regular period mappings (as setup for "default" and "explicit" mappings). **No** is the default option for processing adjustments.
- Yes—If **Yes** is selected, then the regular period and adjustment period are included. If the adjustment period does not exist, then only the regular period is processed.
- Yes (Adjustment Only)—If **Yes (Adjustment Only)** is selected, the system processes the adjustment period only. However, if the adjustment period does not exist, the system pulls the regular period instead.

18. Click **Save**.

Adding Filters for Data Load Rules

For data rules used to import data from the Oracle General Ledger and use filters to limit the results.

Oracle Hyperion Financial Data Quality Management, Enterprise Edition automatically creates filters when a rule is created. You can modify the filters as needed but cannot delete them. (If filters are deleted, FDMEE recreates the default value.)

Data load rule filters:

Oracle General Ledger Dimension	Filter
Scenario	Actual
Balance Amount	Ending Balance
Amount Type	YTD
Currency Type	Total
All Other Dimensions	'@ILvl0Descendants("All ' TARGET_DIMENSION_NAME ' Values")'

Note:

Drill Through is only supported if you load leaf level data for Oracle General Ledger Chart of Account segments. If you load summary level data, then drill through does not work.

To assign a filter to the data load rule:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. Select the data load rule to which to add a filter.
3. Select the **Source Options** tab.

Dimension Name	Filter Condition	Select
Ledger	"All Ledgers].[Vision Foods - USA Ledger"	...
Company	"All Company Values].[3111"	...
Cost_Center	@ILvl0Descendants("All Cost_Center Values")	...
Program	@ILvl0Descendants("All Program Values")	...
Location	@ILvl0Descendants("All Location Values")	...

4. In the **Source Filters** area, click .
5. Select the **Dimension Name**.
6. In **Filter Condition** specify the filter condition:
 - Enter a member name or filter condition in the Filter Condition text box.
 - Click  to display the Member Select screen and select a member using the member selector. Then, click **OK**.

The Member Selector dialog box is displayed. The member selector enables you to view and select members within a dimension. Expand and collapse members within a dimension using the [+] and [-].

The Selector dialog box has two panes—all members in the dimension on the left and selections on the right. The left pane, showing all members available in the dimension, displays the member name and a short description, if available. The right pane, showing selections, displays the member name and the selection type.

You can use the **V** button above each pane to change the columns in the member selector.

Note:

Assign filters for dimension. If you do not assign filters, numbers from the summary members are also retrieved.

To use the member selector:

- a. In the list of available dimensions and members on the left, select a member and click .
- b. To deselect a member from the list of members, click .
- c. To add special options for the member, click  and select an option.

In the member options, “I” indicates inclusive. For example, “IChildren” adds all children for the member, including the selected member, and “IDescendants” adds all the descendants including the selected member. If you select “Children,” the selected member is not included and only its children are included.

The member is moved to the right and displays the option you selected in the Selection Type column. For example, "Descendants" displays in the Selection Type column.

 Tip:

To clear all members from the list of selections, click .

- d. Click **OK** twice to continue defining the source filter details.

The selected member is displayed in Oracle Essbase syntax in the Filter Condition field.

Drilling Through to the Oracle Financials Cloud

Drill though enables you to display and view the account balance summary page in the Oracle Financials Cloud.

When integrating with the Oracle General Ledger, Oracle Hyperion Financial Data Quality Management, Enterprise Edition determines the drill URL definition based on the connection information automatically such as system and fixed information. You do not need to set up when drilling through to the Oracle General Ledger.

Writing Back to the Oracle Financials Cloud

Once budget information is complete in your application, you can define the Oracle Enterprise Performance Management Cloud application as a source and then write back data to the Oracle General Ledger a target. After specifying any necessary filters, you can then extract budget values from EPM Cloud and write them to Oracle General Ledger. In the Export workflow step, the data is written to a flat file, which in turn is copied to a file repository.

For Oracle Planning and Budgeting Cloud users, watch this tutorial video to learn about writing back EPM Cloud budgets to the Oracle General Ledger:



[Tutorial Video](#)

For Oracle Enterprise Planning and Budgeting Cloud users, see the [Tutorial Video](#).

To write back to the Oracle General Ledger:

1. Create an import format to map dimensions to the Oracle General Ledger:
 - a. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
 - b. Click **Add**.
 - c. In **Name**, enter the name of the import format.
 - d. In **Source**, select the name of the EPM application from the drop-down.
 - e. In **Description**, enter a description that can be used to identify the import format.
 - f. In **Drill URL** leave blank.
 - g. Click **Save** to save the import format and see the lower portion populated.

- h. Scroll down to the lower region of the Import Format screen to map EPM Cloud dimensions to the general ledger dimensions.
 - i. Map a source for the target dimension "Ledger."
You can map a dimension like "Entity" to the ledger, and define any necessary data load mapping to convert to the Oracle General Ledger name. If you are writing back to a single ledger, enter the name of the ledger in the expression column.
 - j. Click **Target Options**, select **Budget Name**.
 - k. In **Expression**, leave blank.

If the target is the budget name, enter the value of the accounting scenario that you plan to use.
2. Create a location.
The location is used to execute the transfer of budget amounts to the Oracle General Ledger. The import format is assigned to the location. If you are using multiple import formats, you also need to define multiple locations.
 - a. On the **Setup tab**, under **Integration Setup**, select **Location**.
 - b. Click **Add**.
 - c. In **Name**, enter a name for the location.

The location name is displayed when you initiate the transfer from the EPM application to the Oracle General Ledger.
 - d. In **Import Format**, select the name of the import format you to use during the transfer.



The Source and Target field names are populated automatically based on the import format.

- e. In **Parent Location**, enter the parent assigned to the location.
Parent mapping are used to share mappings with other locations. Enter mappings at the parent location, and the related locations can use the same mappings. Multiple locations can share a parent. This feature is useful when multiple locations use one chart of accounts. Changes to a child or parent mapping table apply to all child and parent locations.
 - f. In **Source**, the source is populated automatically.
 - g. In **Functional Currency**, specify the currency of the location.
 - h. **Optional:** In **Logic Account Group**, specify the logic account group to assign to the location.
 - i. **Optional:** In **Check Entity Group**, specify the check entity group to assign to the location.
 - j. **Optional:** In **Check Rule Group**, specify the check rule group to assign to the location.
 - k. Save the location.
- See [Defining Locations](#).
3. Create **period mappings**.

The period mapping is used to convert periods to Oracle General Ledger accounting calendar periods for the transfer.

**Note:**

When specifying the period, the starting and ending periods should be within a single fiscal year. Providing date ranges that cross fiscal year results in duplicate data.

- a. On the **Setup tab**, under **Integration Setup**, select **Period Mapping**.
- b. Click **Add** and add a separate row for each period that is to receive budget amounts.

Use the period names from the accounting calendar used by the ledger in the general ledger.

- c. Define a **Period Key**.

Once you select a value, information about the period key, prior period key, period name, and the target period month are populated automatically.

- **Target Period Month**—The values in this field need to match the accounting calendar for the ledger in the Oracle General Ledger, which receives the transferred amounts.
- **Target Period Year**—Use values that corresponds to the accounting period (as defined in the Target Period Month column).

See [Defining Period Mappings](#).

4. Define the **data load rule**.

A data load rule is used to submit the process to transfer balances from the EPM application to the Oracle General Ledger. The data load rule is created once, but used each time there is a transfer.

- a. On the **Workflow tab**, under **Data Load**, select **Data Load Rule**.
- b. From the **POV Bar**, select the location to use for the data load rule.

Data load rules are processed within the context of a point of view. The default point of view is selected automatically. The information for the point of view is shown in the POV bar at the bottom of the screen.
- c. Click **Add**.
- d. In **Name**, enter the name of the data load rule.
- e. In **Category**, leave the default category value.
- f. In **Description**, enter a description to identify the data load rule when you launch the request to transfer general ledger balances.
- g. In **Target Plan Type**, select a plan type.
- h. In **Period Mapping Type**, select the period mapping type for each data rule.

Valid options:

- **Default**—The Data Rule uses the Period Key and Prior Period Key defined in Oracle Hyperion Financial Data Quality Management, Enterprise Edition

to determine the Source General Ledger Periods mapped to each FDMEE period included in a Data Rule execution.

- **Explicit**—The Data Rule uses the Explicit period mappings defined in FDMEE to determine the source GL Periods mapped to each FDMEE Period included in a Data Rule execution. Explicit period mappings enable support of additional GL data sources where periods are not defined by start and end dates.
- i. Click **Save**.
5. Add **Source Option filters** to the data load rule for write-back.
- a. On the **Workflow tab**, under **Data Load**, select **Data Load Rule**.
 - b. From the **POV Bar**, select the location to use for the data load rule.
- Data load rules are processed within the context of a point of view. The default point of view is selected automatically. The information for the point of view is shown in the POV bar at the bottom of the screen.
- c. Select the data load rule to which to add a filter.
 - d. Select the **Source Options** tab.
 - e. In the **Source Filters** area, click .
 - f. Select the **Dimension Name**.
 - g. In **Filter Condition** specify the filter condition:
 - Enter a member name or filter condition in the Filter Condition text box.
 - Click  to display the Member Select screen and use a member selector to specify functions for filtering. Then, click **OK**.

To use the member selector:

- i. In the list of available dimensions and members on the left, select a member and click .
 - ii. To deselect a member from the list of members, click .
 - iii. To add special options for the member, click  and select an option.
- In the member options, “I” indicates inclusive. For example, “IChildren” adds all children for the member, including the selected member, and “IDescendants” adds all the descendants including the selected member. If you select “Children,” the selected member is not included and only its children are included.

The member is moved to the right and displays the option you selected in the Selection Type column. For example, “Descendants” displays in the Selection Type column.



Tip:

To clear all members from the list of selections, click .

- iv. Click **OK** twice to continue defining the source filter details.

The selected member is displayed in Oracle Essbase syntax in the Filter Condition field.

6. Execute the data load rule to write back.
 - a. On the **Workflow tab**, under **Data Load**, select **Data Load Rule**.
 - b. From the **POV Bar**, verify the location and period to use for the data load rule.
 - c. Select **Execute** to submit a request to transfer budget amounts to the Oracle General Ledger.
 - d. In **Import from Source**, select to import the budget information from Oracle Hyperion Planning.
 - e. In **Recalculate**, leave blank.
 - f. In **Export to Target**, select to export the information to the Oracle General Ledger.
 - g. In **Start Period**, select the earliest general ledger period to transfer.

The list of values includes all the general ledger periods that you have defined in the period mapping. This is typically the first period of the year for the initial budget load, and then the current period or a future period during the year if there are updates to the budget that are to be transferred to the Oracle General Ledger.
 - h. In **End Period**, select the latest General Ledger period to transfer.

The list of values includes all the general ledger periods you have defined in the period mapping.
 - i. In **Import Mode**, select **Replace** to overwrite existing budget information in Oracle General Ledger for the period range you selected (from the start period and end period options).

Select **Append** to add information to existing Oracle General Ledger budget amounts without overwriting existing amounts.
 - j. Click **Run**.

Extracting the Integrated EPM Cloud and Oracle General Ledger Data to a Flat File

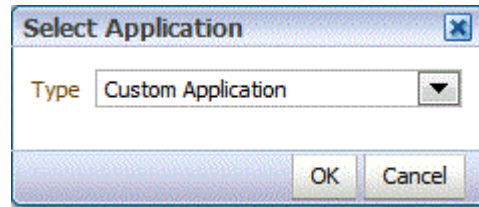
After completing a planning or budgeting cycle in the EPM application, data can be written out to a text file for use in an external application. Exporting data from the EPM application is facilitated through the use of a custom application.

If multiple charts of accounts are used in the Oracle General Ledger and they need to be mapped differently, create a separate target application for each chart of accounts structure. It may be useful to include the chart of accounts name as part of the target application to identify the target applications when completing later configurations.

To extract the data from the integrated file:

1. Create a custom target application.
 - a. On the **Setup tab**, under **Register**, select **Target Application**.
 - b. In **Target Application**, in the **Target Application** summary grid, click **Add**.

- c. From **Select Application**, select **Custom Application**, and then click **OK**.



- d. In **Application Details**, enter the **application name**.

 Note:

Do not use spaces between words in the name or the transfer will fail.

- e. Select the **Dimension Details** tab.

- f. Specify the **Dimension Name**.

Oracle Hyperion Planning uses dimensions to create and maintain budget information. These dimensions must be mapped to the EPM application. This enables you to transfer the budget information in a manner recognized by the EPM application. Include the dimensions in the custom application that you want in your output file. You then map the EPM application dimensions to the dimensions in the custom application.

Create a row for each EPM application dimension that should be mapped to the Oracle General Ledger.

The following dimension names must be mapped as follows:

- Budget Name must be mapped to the target dimension class "Account." Data Table Column Name must be "Account." This is distinct from mapping the account field of the chart of accounts, which must be done separately.

This is also distinct from mapping the account field of the chart of accounts, which must be done separately.

- Ledger must be mapped to the Target Dimension Class Entity. The data table column name must be ENTITY.

Additionally, there should be a mapping from the EPM application for each segment of the chart of accounts.

- g. In **Target Dimension Class**, use the value "Generic" for all rows except Account and Entity, as described under Dimension Name.
- h. In **Data Table Column Name**, use the values UD1, UD2, UD3 based upon the order of the segments from the chart of accounts definition.

UD stands for "User Defined."

For example, a chart of accounts structure has three segments defined in this order: Company, Cost Center, and Account, and the values for the data table column names would be:

- "Company" uses the data table column name UD1.
- "Cost Center" uses the data table column name UD2.
- "Account" uses the data table column name UD3.

If there is a fourth segment of the chart such as Subaccount, map it to UD4.

- In **Sequence**, leave blank.
- Click **Application Options**.

Property Name	Value	Select
Enable export to file	Yes	
File Character Set	UTF-8	
Column Delimiter	,	
File Name for Download		

- In **Enable export to file**, select **Yes** to have Oracle Hyperion Financial Data Quality Management, Enterprise Edition create an output data file for the custom target application.

As part of the transfer, FDMEE calls the General Ledger Validate and Upload Budgets.

A file is created in the outbox folder on the server. For example when the location is named "Texas" and the next sequence is 16, then the file name is Texas_15.dat. The file is created during the export step of the workflow process.

- In **File Character Set**, select the file character set.

The file character set determines the method for mapping bit combinations to characters for creating, storing, and displaying text. Each encoding has a name; for example, UTF-8. Within an encoding, each character maps to a specific bit combination; for example, in UTF-8, uppercase A maps to HEX41.

- In **Column Delimiter**, select the character to use for delimiting columns in the output file.

Available column delimiters are:

- ,
- |
- !
- ;
- :

2. Create an import format.

- On the **Setup** tab, under **Integration Setup**, select **Import Format**.
- Click **Add**.
- In **Name**, enter the name of the import format.
- In **Source**, enter the name of the Planning application.

- e. In **Description**, enter a description that can be used to identify the import format.
 - f. In **Drill URL** leave blank.
 - g. In **Target Type**, select **EPM**.
 - h. In **Target**, enter the name of the custom target application that you defined for the transfer.
 - i. Scroll down to the lower region of the Import Format screen to map the EPM dimensions.
 - j. In **Source Column**, select the appropriate values from the EPM application as follows:

To map the Budget Name, and the natural account segment of the chart of accounts, select "Account."

To map the Ledger and chart of accounts segment that are based upon the entity (such as the company or department), select "Entity."
 - k. In **Expression**, leave blank.

If the target is the budget name, enter the value of the accounting scenario that you plan to use.
3. Define locations.

See [Defining Locations](#).
 4. Enter the General Ledger Period name in the Target Period Month to define the Application Period mapping.

See [Application Mappings](#).
 5. Define data load mappings by mapping the Ledger, Budget, and Segments.

See [Creating Member Mappings](#).
 6. Define the data load rule.
 7. Execute the data load rule.

See [Running Data Load Rules](#).

When you run the data load rule, the output file is created and can then be downloaded. You go to the process details screen, find the job that you just executed and scroll to the far right until you see the download link, and click on this link to download the file that was just created.
 8. On the **Workflow tab**, under **Monitor**, select **Process Details**.
 9. Select the job that you just executed and scroll to the far right until you see the download link.
 10. Click the link to download the file that was just created.

Using Excel Trial Balance Files to Import Data

An Excel trial-balance file is an Excel spreadsheet that, through the import screen, is formatted to one more periods, categories, and locations.

Text Trial Balance Files Versus Excel Trial Balance Files

Text trial-balance files and Excel trial-balance files are similar in two ways: They are both loaded to the current POV (category and period), and, on the import form, they use the same Append and Replace options.

Text trial-balance files and Excel trial-balance files differ in one way: text files can use only the standard import format, but Excel data files do not use any import formats.

When you use an Excel Trial Balance template, the template can contain one or more periods. When doing a multiple period load, you need to create a dummy import format that indicates multiple periods. If just a single data value is in the Excel file, then you don't need an import format.

Downloading an Excel Trial Balance Template

To download an Excel trial balance template:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Workbench**.
2. From the **Download Template** drop-down, select **Trial Balance**.
3. From the **Open** screen, open or save the template and click **OK**.

Defining Excel Trial Balance Templates

To define an Excel trial-balance template, you define the first row of the named region, which contains the metadata tags. Other information may be defined outside of the region, but the system only considers what is within the named region. For example, the template includes a title and an amount summary. These are outside the named region and are not processed when loading data.

To load data using a template, the system uses a named range definition to find the dimensions and the related data. For the trial balance template, the predefined range is called upsTB, and it can be seen using the "Name Manager" option in Excel.

The following template contains one line of metadata (row 1) and three lines of imported data (rows 5–7).

Dimension Values and Amount should be populated in the respective columns as per the Tags defined in row 1. To add additional dimension tags, add columns. Add data by adding rows.

When adding rows or columns, add them within the named region. Excel updates the region definition automatically. If you add rows outside of the region, update the region to include these new rows or columns. When adding dimension columns, add a dimension tag to specify when the column is an account, entity, intercompany transaction, amount or user defined (UD) dimension. Note that the entity dimension is represented by the tag for "Center."

Table 3-14 FDMEE dimension tags and the corresponding tags

Oracle Hyperion Financial Data Quality Management, Enterprise Edition Dimension	Valid Tags
Account (Required)	A, Account, SrcAcctKey

Table 3-14 (Cont.) FDMEE dimension tags and the corresponding tags

Oracle Hyperion Financial Data Quality Management, Enterprise Edition Dimension	Valid Tags
Center (Required)	C, Center, SrcCenterKey
Description (Optional)	D, Description, SrcAcctDesc
IC Counter Party (Optional)	I, IC, ICCoParty
User Defined 1-User Defined 20 (Optional)	1-20, UD1-UD20, UserDefined1- UserDefined20
Amount (Required)	V, Amount, SrcAmount

In the template that is provided with FDMEE, some of the rows are hidden. To update the columns and the column tags, you need to unhide these rows. To do this, select the row above and below the hidden rows, and then update the cell height. A setting of 12.75 is the standard height for cells, and this should show all of the hidden rows for the selected range in the sheet. You may want to re-hide the rows after you have made the necessary changes.

	A	B	C	D
1	Trial Balance Template			
2				
3				
4				
5	Account	Center	Description	Current Month
8				
9				
10				
11				

Adding a Multiple Period Data Load Using Excel

You may also use the Excel Trial Balance template to load data to multiple periods. To do this, create a data rule using a multiple period import format. The import format does not have to contain any detail mappings, only the definition must have a multiple period. Using the multiple period data rule you can import the Excel Trial Balance File. You create a dummy import format and only select the specification for the multiple periods. To load data for multiple periods, the column header must be in the format V1:PeriodKey, V2:Periodkey, etc. The period key must be specified in YYYY/MM/DD format. You do not have to define the source period mapping in the data rule. You also need to update the range to make sure any additional columns are included in the range. Below is a sample of an Excel file.



Note:

You only need to include a period key (for example, V1:2016/1/31) with the tag if the periods are non-contiguous. If contiguous periods, then the period keys are ignored and the start/end selected when running the rule is used to define the periods.

 Note:

The Excel template expects an empty row between the tags and the first row of data.

5	Account	Center	Description	Jan	Feb
6	A	C	D	V1:2013/1/1	V2:2013/3/1
8	Revenue		100	110	1,300
9	Expense		100	500	500
10					

Importing Excel Mapping

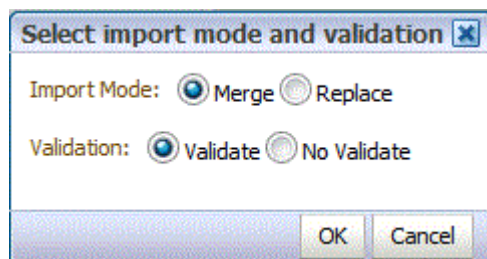
You can import Excel mappings by selecting the Import option and selecting an Excel mapping.

 Note:

The import of mapping rules using an Excel template provides a place to specify a mapping script.

To import an Excel mapping:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Mapping**.
2. Select the **All Mapping** tab.
3. From the **Import** drop-down, select **Import from Excel**.
4. From **Select a file to import**, select the Excel file to import, and then click **OK**.
5. From **Select mode and validation**, in **Import mode**, select the import mode.



- Merge—Overwrites the data in the application with the data in the Excel data load file.
 - Replace—Clears values from dimensions in the Excel data load file and replaces them with values in the existing file.
6. Click **Validate** to validate the mappings.

7. Click **OK**.

The mapping inherits the default data load rule, and the description of "System Generated Mappings."

Using Journal Templates to Import Data

In Oracle Hyperion Financial Management, you use journals to adjust data after it has been entered or loaded into base level entities. Journals provide an audit trail of changes made in the application and indicate, which users made adjustments and which accounts, entities, and time periods are affected.

Oracle Hyperion Financial Data Quality Management, Enterprise Edition enables you to load Financial Management journal entries with journal templates. These templates are Excel spreadsheets that are formatted as journal entry input screens.

FDMEET journal templates are typically used for the following types of adjustments:

- GAAP adjustments to general ledger files
- Transformations of gross balance accounts into roll-forward accounts (Beg, Add, Del, End)
- Supplemental data entries (Head Count, Ratios, and so on)

Additional Considerations for Loading Journal Templates

Note the following when loading journal templates:

1. Journals may only be loaded manually by way of the Oracle Hyperion Financial Data Quality Management, Enterprise Edition user interface. You cannot load journals in offline or batch mode.
2. The Oracle Hyperion Financial Management journal group is not supported, only the journal label. The journal ID from the journal template is used as the journal label when loading to Financial Management, and the group is left blank.
3. Only one description per journal is loaded, and the load process uses the last description it finds as the description for the journal.
4. The UpCheck feature that was available in the legacy Oracle Hyperion Financial Data Quality Management product for journal data validation is not available in FDMEET. Users can provide similar functionality by using a custom event script. See [Using Event Scripts](#).

Integrating Financial Management Journals

Integrating Oracle Hyperion Financial Management journal feature with Oracle Hyperion Financial Data Quality Management, Enterprise Edition involves setting up the FDMEET application options, and integrating data values.

Financial Management Application Options

The following Oracle Hyperion Financial Management application options must be configured before using the Journal feature in Oracle Hyperion Financial Data Quality Management, Enterprise Edition:

- Enable Journal Load

- Journal Enable JV ID per Entity
- Journal Balancing Attribute
- Journal Status

For information on setting up any of the above options, see [#unique_124](#).

Data Values

Data value is an extra dimension that is only used when integrating with an Oracle Hyperion Financial Management multi-dimension target system. The name of the dimension is "Value." The members in this dimension are: [Contribution Adjs], and [Parent Adjs]. When data is loaded to Financial Management, specify a member of the value dimension to indicate where the data is loaded. In the Location definition in Oracle Hyperion Financial Data Quality Management, Enterprise Edition, specify an entry for the value dimension in the Data Value field. The Data Value is set on the Location screen by selecting the Search link.

When FDMEE creates the load file, this dimension value is entered for every data line loaded by this location. You must enter a value in this field to integrate with Financial Management, or else the validation fails. The default value is Data Value <Entity Currency>.

If you load journals to Financial Management, you can specify the value dimension member for data loads and for journal loads. The first ";" is the value member used for data loads, and the second field by ";" is the value member for journal loads.

When using the template, the system picks up the value member by looking for the second field delimited by ";" in the value member field in the location

When Search is selected, FDMEE connects to the Financial Management to get a list of valid data values. FDMEE takes the values from Financial Management and adds rows created by FDMEE that are a concatenation of the original value and "Adjustment Data Values". FDMEE uses these newly created rows to manage journal loading to Financial Management.

The rows that FDMEE creates in the Data Value selection screen are:

- [Contribution Adjs];[Contribution Adjs]
- [Contribution Adjs];[Parent Adjs]
- [Contribution Adjs];<Entity Curr Adjs>
- [Contribution Adjs];<Parent Curr Adjs>
- [Parent Adjs];[Contribution Adjs]
- [Parent Adjs];[Parent Adjs]
- [Parent Adjs];<Entity Curr Adjs>
- [Parent Adjs];<Parent Curr Adjs>
- <Entity Curr Adjs>;[Contribution Adjs]
- <Entity Curr Adjs>;[Parent Adjs]
- <Entity Curr Adjs>;<Entity Curr Adjs>
- <Entity Curr Adjs>;<Parent Curr Adjs>

Downloading a Journal Template

If you use a journal template to load journal entries, then from the Data Load Workbench you can select a journal template that has been uploaded to the server. A template is associated with an existing POV. Once you have completed the required values for the template, you reload it (post it to the server), create a new data load rule for that POV, and specify a load type of "Journal" in the data load rule.

To download a Journal template:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Workbench**.
2. From the POV, select the POV associated with the journal template to download.
3. In **Download Template**, select **Journal**.
4. On the **Open** screen, open or save the template, and then click **OK**.

Defining Journal Templates

You define an Oracle Hyperion Financial Data Quality Management, Enterprise Edition journal template by specifying the metadata header that FDMEE uses to interpret the data contained in the template. Metadata consists of a series of tags that tell FDMEE what column contains the account number or what period in which to load. A named range is used to tell the system where to look for the header information and the data to load. For the journal template, the named region is called upsJournal, and the following details relate to the relative rows within the named range. The template that comes with FDMEE has a range that starts at row 16, but the metadata starts at row 1 within the range. See below for an example of the template that is provided with FDMEE.

The sample journal template below has two lines of actual imported data, and five lines of metadata. Rows 1-5 contain metadata and rows 6 and 7 contain the data values.

	A	B	C	D
1	ID-Texas100		1 Texas	
2			ACTUAL05	
3			1/31/2005	
4			A	
5	A	C	V	D
6	1100	Texas	500.00	Reclass Cash
7	1210	Texas	(500.00)	Reclass Cash

Metadata Structure

The metadata header (Row 1-5) instructs Oracle Hyperion Financial Data Quality Management, Enterprise Edition on how to find the relevant segments of data that it handles in this template. The following Row 1-5 topics explain how each piece of metadata is used by FDMEE.

Row 1 (Journal ID and Location Tag)

The tag in row 1 of the range is used to set the Journal ID and the Oracle Hyperion Financial Data Quality Management, Enterprise Edition location that the data should

be loaded into. The Journal ID must be placed in row 1 of the Account column. Place the Location tag in row 1 of the Amount column.

Row 2 (FDMEE Category Tag)

The tag in row 2 of the range sets the Oracle Hyperion Financial Data Quality Management, Enterprise Edition category into which the journal loads. The category must be a valid FDMEE category. The FDMEE Category tag must be placed in the Amount column.

Row 3 (FDMEE Period Tag)

The tag in row 3 of the range sets the period that the data should be loaded into. The period must be a valid Oracle Hyperion Financial Data Quality Management, Enterprise Edition period. This tag must be placed in the Amount column.

Row 4 (Load Method Tag)

The tag in row 4 of the range sets the journal load method within Oracle Hyperion Financial Data Quality Management, Enterprise Edition. To append to an existing journal with the same Journal ID, type the letter **A**. If a journal exists with the same Journal ID within the same FDMEE point of view, then the new journal is appended to the previously submitted journal. To replace an existing journal with the same Journal ID, type the letter **R**.

If a journal exists with the same journal ID within the same FDMEE point of view, then the new journal replaces the old journal. This tag must be placed in the Amount column. The table below defines the possible tags. The "AZ" and "RZ" settings work the same as the "A" and "R" methods except that all zero value amounts are suppressed.

Table 3-15 Journal Load Methods and Corresponding Tags

Method	Valid Tags
Append Journal	A, Append
Replace Journal (default)	R Replace
Append journal - Zero Suppress	AZ
Replace journal - Zero Suppress	RZ

Row 5 (Dimension Tags)

The tags in row five define the dimension that the amounts are loaded into. The table below defines the possible dimension tags.

Table 3-16 Journal Load Method and Corresponding Tags

Oracle Hyperion Financial Data Quality Management, Enterprise Edition Dimension	Valid Tags
Account (Required)	A, Account, SrcAcctKey
Center (Required)	C, Center, SrcCenterKey
Description (Optional)	D, Description, SrcAcctDesc

Table 3-16 (Cont.) Journal Load Method and Corresponding Tags

Oracle Hyperion Financial Data Quality Management, Enterprise Edition Dimension	Valid Tags
IC Counter Party (Optional)	I, IC, ICCoParty
User Defined 1-User Defined 20 (Optional)	1-20, UD1-UD20, UserDefined1-UserDefined20
Amount (Required)	V, Amount, SrcAmount

Creating Range Names Within Journal Templates

A range of columns comes predefined in the template, and you can simply fill out the top section in rows 5-13 and add any necessary columns. Oracle Hyperion Financial Data Quality Management, Enterprise Edition evaluates the named range to determine the POV, load method, and data to load.

To create your own journal template, you create a range name that includes all metadata and data cells, and that begin with the prefix *ups*. For example, for a standard template, create the range name [upsStandardJV (B16 to J33)].

The following illustration depicts a journal template. Note that in this template, the metadata are not in rows 1–5, but in rows 16–20. The template has an upsJournal starting from row 16. Therefore, rows 16–20 are the first five rows in the upsJournal. Rows 4–14 is a simple interface to assist users with creating the metadata header. Metadata information is entered here, and referenced by the metadata header.

(Enter journal data against the respective columns and by adding more rows within the range. The easiest thing to do is to add rows to the existing range and just use a single range, and use the default upsJournal. You add columns to the spreadsheet based on the dimensionality of the target application.)

A	B	C	D	E	F	G	H	I	J
1	Journal Details								24-Oct-16
2									
3									
4									
5	Journal ID:	JL_WestReg	Enter the Journal ID. It must be 10 or less characters						
6	Location:	FCCSAPP1	Enter the Location Name. It must match the value in POV in the User Interface						
7	Category:	Actual	Enter the Category Name. It must match the value in POV in the User Interface						
8	Period:	12/16/2016	Enter the Period Name. It must match the value in POV in the User Interface						
9	Load Method:	R	A = Append, R = Replace						
10									
11									
12									
13									
14									
15	Account	Entity	Intercompany	Movement	Multi GAAP	Custom1	Custom2	Amount	Description
16	JL_WestReg							FCCSAPP1	
17								Actual	
18								12/16/2016	
19								R	
20	A	C	I	3	4	1	5	V	D
21	1100	E1						1,000.00	Reclass Cash
22	1111	E1						(1,000.00)	Reclass Cash
23									
24									
25									
26									
27									
28									
29									
30									
31									
32									
33									
34									

The journal template must contain the following dimensions:

- **Account**—It is the journal account (required).
- **Entity**—It is the journal entity (required).
- **Intercompany**— (Optional unless being used by Oracle Financial Consolidation and Close Cloud.)

- **Movement**—Data changes from period to period (required)
- **Multi GAAP**—Data changes from period to period (optional)
- **Custom1, Custom2 Dimensions**—Optional unless used by the target application
- **Amount**—(required)
- **Description**—(optional)

Processing Journals

The process for processing journals is:

1. Load the journal file in Excel format from the `inbox` directory.
2. Check whether the POV entered in the journal matches the current POV in Oracle Hyperion Financial Data Quality Management, Enterprise Edition. The ups range is also checked.

When a journal is checked in, FDMEE examines the template for all ranges with names beginning with ups. It then examines and validates the metadata tags found in each ups range. FDMEE does not check in metadata segments that include an invalid range.

3. Post the journal.

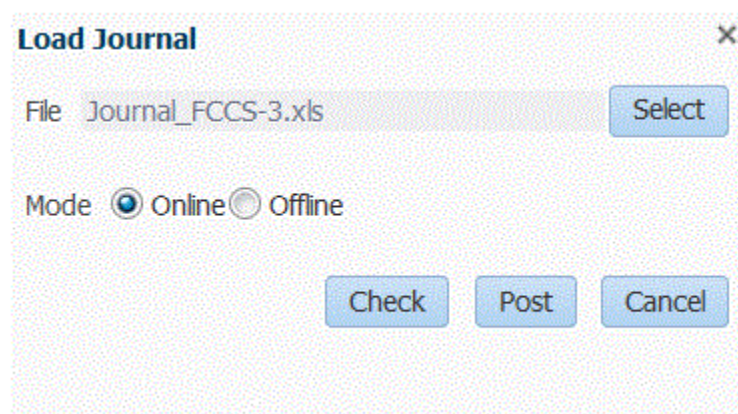
Loading Journals

To load a journal:

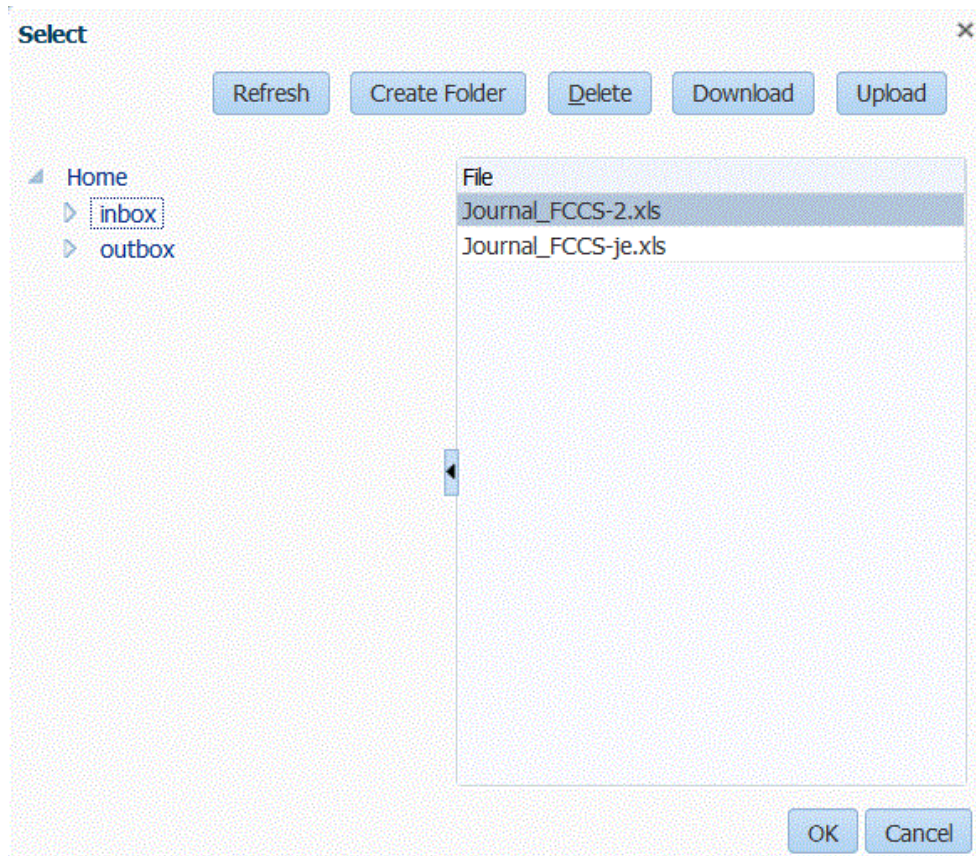
1. On the **Workflow** tab, under **Data Load**, select **Data Load Workbench**.

When you load a journal, Oracle Hyperion Financial Data Quality Management, Enterprise Edition uses the current POV to determine location, category, and period. To use another POV, select another POV on the Data Load Workbench.

2. Click **Load Journal**.



3. On the **Load Journal** screen, to browse for a journal file, click **Select**.



- a. Select a journal template to load from the server to which you uploaded one and click **OK**.

When a journal has been successfully loaded, the **Check** button is enabled.

 Note:

When loading journals to an Oracle Financial Consolidation and Close Cloud target, consider that FDMEE does not determine the account types or select the credits/debits. All positive numbers are loaded as credits and all negative numbers are loaded as debits. If you need to designate other credit or debit signs for your account type, use the change sign feature in Data Load Mappings or another customized method to handle credits/debits changes for your journal loads.

- b. **Optional:** To download a journal file, click **Download** and open or save the journal file.
- c. **Optional:** To upload a journal file, click **Upload**, then navigate to the file to upload, and click **OK**.
4. Click **Check** to validate and load the journal.
See [Checking Journals](#).

Checking Journals

Before journals can be posted, they must be checked. This process verifies whether the POV entered in the Excel file for the journal matches the current POV. It also ensures that the ups range is valid. If the validation is successful, the Post button is enabled.

 Note:

If the journal import file is not XLS or XLSX, then the check feature is not available.

To check a journal:

1. Make sure that a successfully loaded journal file is in the **File** field.
The journal file must be an Excel (.xls) file type.
2. Click **Check**.
3. Select **Online** or **Offline** for the processing method.

Online checking runs immediately, and offline checking runs in the background.

When a journal is checked, Oracle Hyperion Financial Data Quality Management, Enterprise Edition examines the journal file for all ranges with names beginning with ups. It then examines and validates the metadata tags found in each ups range. FDMEE does not check metadata segments that include an invalid range.

When FDMEE validates the journal, you get the following message: "The journal file checked successfully."

Posting Journals

After a journal has been validated (checked) successfully, you can post the journal. Posting a journal appends or replaces the data displayed in the Import Format screen (as determined by the load method specified in the journal).

To post the journal:

1. Select the journal.
2. Click **Post**.

When Oracle Hyperion Financial Data Quality Management, Enterprise Edition posts the journal, you get the following message: "The journal file loaded successfully."

Journal Security

If the POV Lock option is enabled, Oracle Hyperion Financial Data Quality Management, Enterprise Edition administrators and end users are restricted to posting journals to the FDMEE global POV.

Loading Data Using a Universal Data Adapter

The Universal Data Adapter enables you to integrate external source table/view information directly into Oracle Hyperion Financial Data Quality Management, Enterprise Edition bypassing the open interface table. This feature allows the user to easily connect to any source data where they have access to the underlying database in the case where a prepackaged adapter is not available, or if the pre-packaged adapter does not provide the required data. This feature effectively provides direct integration to any source table or view.

At a high level, the process for integrating a Universal Data Adapter with FDMEE is:

1. In Oracle Database Integrator, create a data server and physical schema for the source system.
2. Create a logical schema for the appropriate technology of the adapter in ODI and map it to the physical schema.
3. In FDMEE, register the source system.
4. Create a source adapter.
5. Register the target application (as an Oracle Enterprise Performance Management System Application).
6. Create an import format (map the source table/view columns and target dimensions). Then regenerate the ODI Scenario.
7. Create the location.
8. Create the period mapping.
9. Create the category mapping.
10. Create the data load mapping.

Defining a Universal Data Adapter in Oracle Database Integrator (ODI)

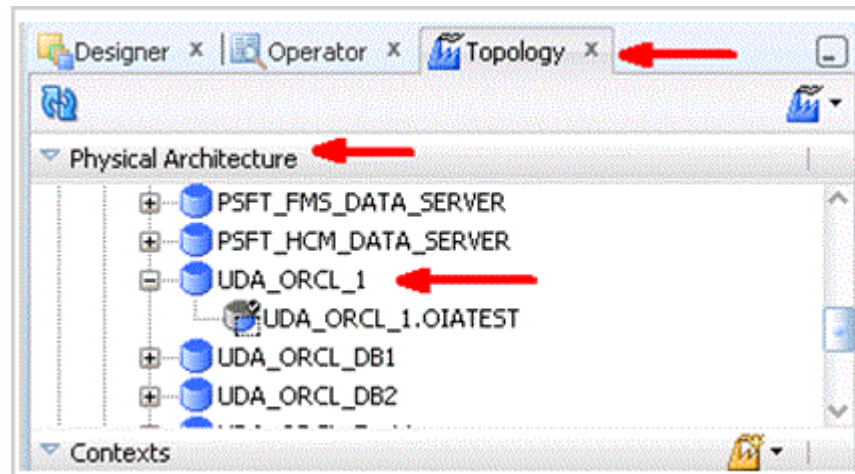
Universal data adapters depend on Oracle Data Integrator (ODI) to import, transform, and validate end export data into target applications. As such, you need to specify how the data is seeded to Oracle Hyperion Financial Data Quality Management, Enterprise Edition in ODI.

Create a Data Server and Physical Schema for the Universal Data Adapter Source

This section describes how to create a data server and a physical schema for universal data adapters provided by Oracle Hyperion Financial Data Quality Management, Enterprise Edition (Oracle, MSSQL Server, MySQL, Teradata, and DB2).

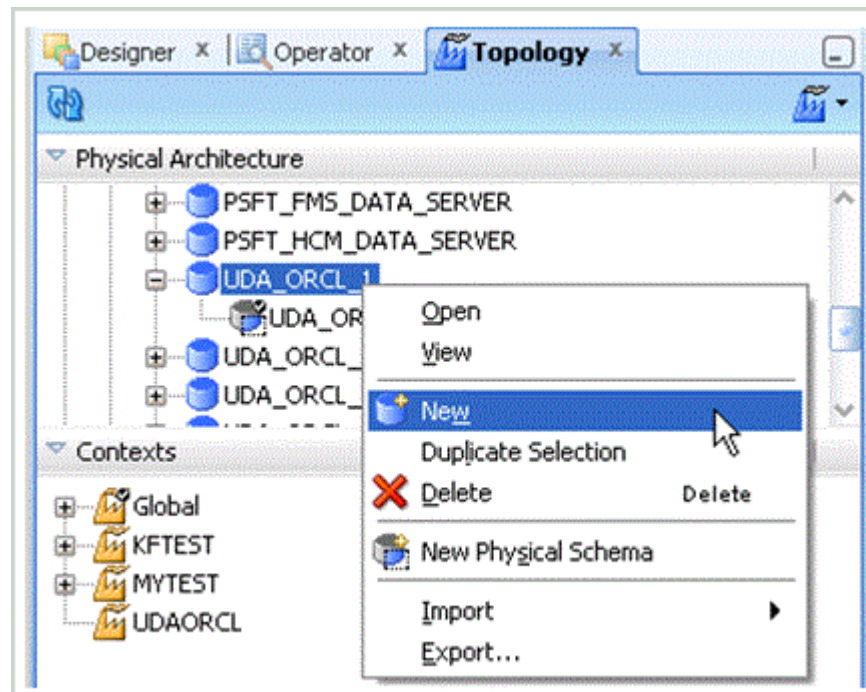
To create a data server and physical schema for the Universal Data Adapter source:

1. Start the **Oracle Data Integrator Studio** (ODI).
2. From **Topology**, then **Physical Architecture**, and then **Technologies**, select the technology of the data server.

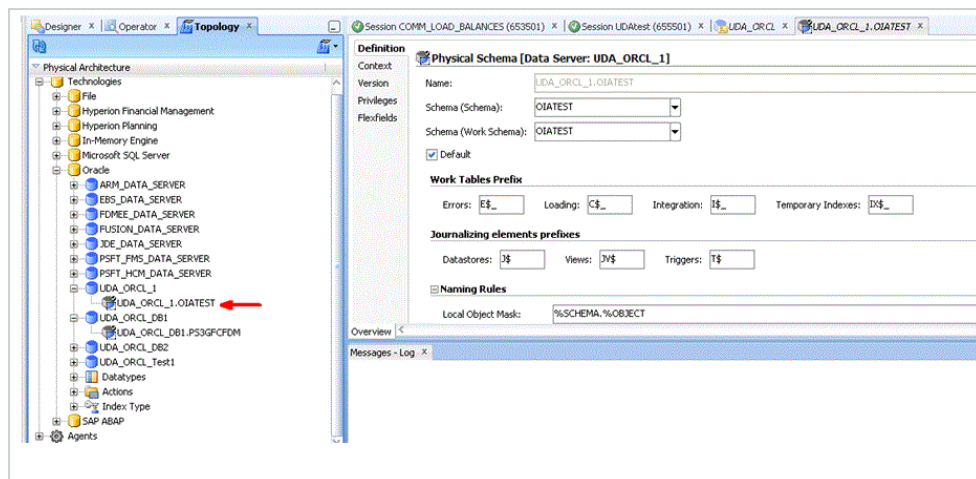
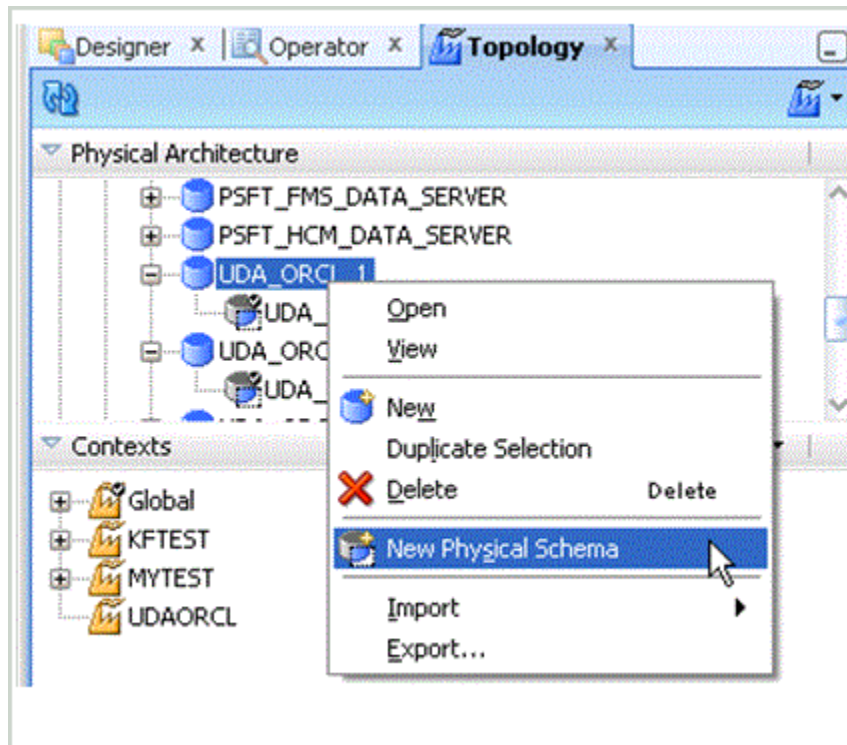


For example, select **Oracle** or **Microsoft SQL Server**.

3. Right click the data server, and then select **New**.



4. Enter the **JDBC** details for the external database.
5. Right click the data server created in step 3, and then select **New Physical Schema**.

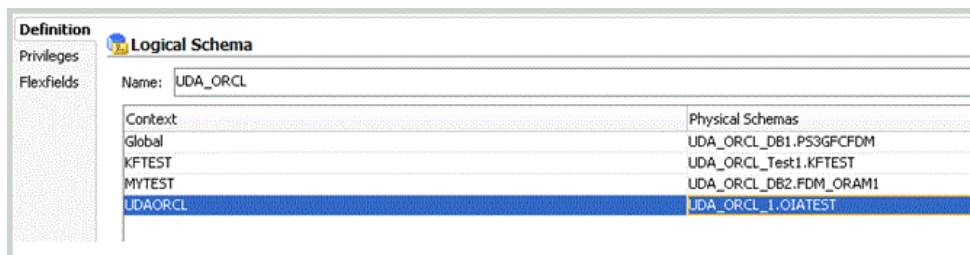
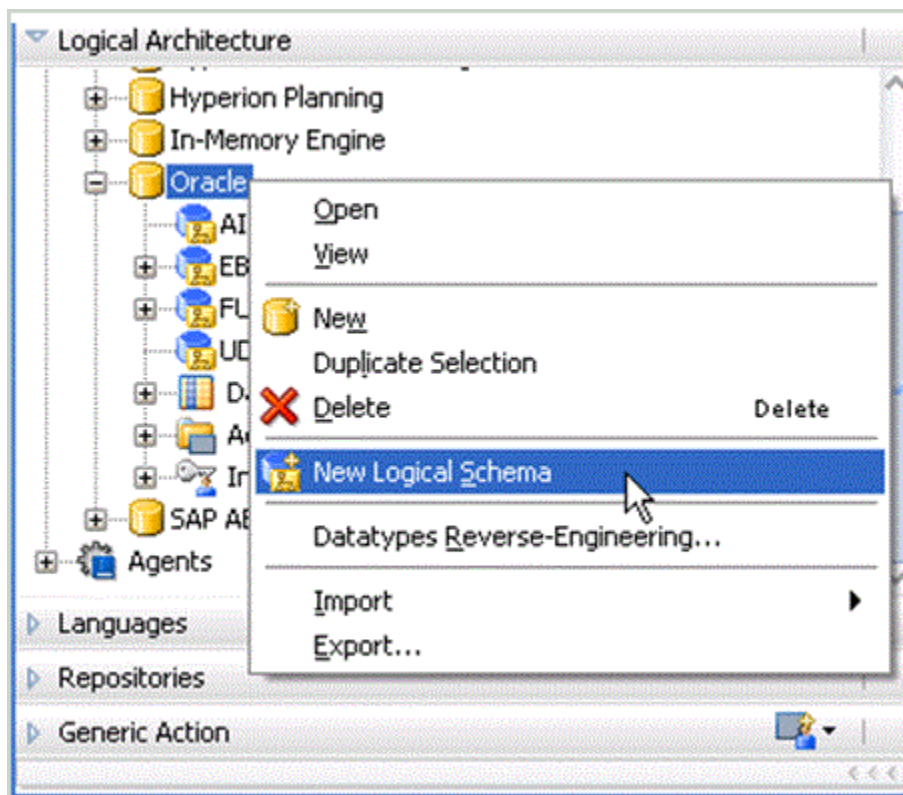


6. From **Topology**, then **Logical Architecture**, and then **Technologies**, select the technology for the logical schema.

Name each logical schema as shown below for each of the database technologies:

- Universal Data (Oracle)—UDA_ORCL
- Universal Data (SQL Server)—UDA_MSSQL
- Universal Data (MySQL)—UDA_MYSQL
- Universal Data (Teradata)—UDA_TD
- Universal Data (DB2)—UDA_UBD

- Universal Data (DB2 400)—UDA_DB2_400
- Universal Data (SAP HANA)—UDA_HANA



Configuring SAP HANA

The following steps are required to use SAP HANA with Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

To set up SAP HANA:

1. Download the `jdbc.jar` used to connect to SAP HANA (`njdbc.jar`) to the target location: `\Oracle\Middleware\user_projects\domains\EPMSys\lib`.
The `jdbc.jar` file is provided by SAP.
2. Import the SAP HANA technology from the: `<EPM_ORACLE_HOME>\products\FinancialDataQuality\odi\11.1.2.4.00\masterrep` folder.

3. Create the SAP Model in the model folder " Universal Data Adapter Model" with the following values:
 - **Name**—HANA source
 - **Code**—HANA_SOURCE
 - **Technology**—SAP HANA
 - **Logical Schema**—UDA_HANA

 Note:

If the model folder "Universal Data Adapter Model" is not available, import it from the following location: <EPM_ORACLE_HOME>\products
\FinancialDataQuality\odi\11.1.2.4.00\workrep.

4. Import the HANA Adapter Project.
5. Create a connection to HANA, and map the logical schema UDA_HANA to it in the context.

Working with a Universal Data Adapter in FDMEE

These sections explain how to configure and use universal data adapters in Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

Working with Universal Data Adapter Source Systems

Before you use the universal data adapter in Oracle Hyperion Financial Data Quality Management, Enterprise Edition, register the data source from which you want to import data.

FDMEE supports data extraction from the following data sources:

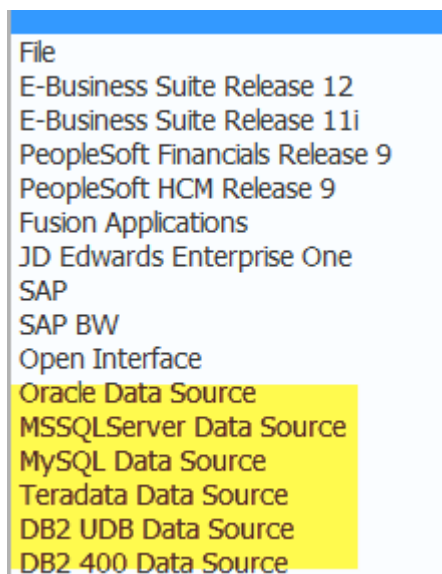
- Oracle Data Source – UDA_ORCL
- MSSQL Server Data Source – UDA_MSSQL
- MySQL Data Source – UDA_MYSQL
- Teradata Data Source – UDA_TD
- DB2 Data Source – UDA_UDB
- DB2 400 Data Source - UDA_DB2/400
- SAP Hana – UDA_HANA

To add a source system for a universal data adapter:

1. On the **Setup** tab, under **Register**, select **Source System**.
2. In **Source System**, click **Add**.
3. Enter the source system details:
 - a. In **Source System Name**, enter the source system name.

Enter the name you want to use for the Data Source, such as "Oracle Receivables Data" or "MYSQL Payables Data", which indicates the specific data source.

- b. In **Source System Description**, enter a description of the source system.
- c. In **Source System Type**, select the type of source system of the universal data adapter.



- d. In **ODI Context Code**, specify the context code defined in ODI for the connection to the specific instance.

The ODI context code refers to the context defined in Oracle Data Integrator. A context groups the source and target connection information.

When data is sourced from another instance of a same seeded technology (Oracle, MSSQL, MYSQL, Teradata, DB2, or DB2 400), then another source system of same technology type with a **different Context**. In the Context, they should map the logical schema belonging to the source system type to the other instance.

4. Click **Save**.

After you add a source system, select the source system in the summary region, and the details are displayed in the lower pane.

Working with Universal Data Adapter Source Systems and Target Applications

The Oracle Hyperion Financial Data Quality Management, Enterprise Edition Target Application feature requires no additional steps when working with universal data adapter source systems.

Creating the Source Adapter

A source adapter is an integration framework in Oracle Hyperion Financial Data Quality Management, Enterprise Edition that enables you to extract data from the source system in a flexible and customizable manner. When you configure Oracle Data Integrator with a universal source adapter, ODI extracts the data directly to a temporary data table used for transforming data from source to target in a data rule execution (tdataseg_t).

As another step in implementing the universal data adapter feature, create a definition that instructs how data is populated in FDMEE from the source table through ODI.

For the source adapter, FDMEE seeds and sets up the source adapter for the following six data sources:

- Oracle
- MSSQL
- MYSQL
- Teradata
- DB2
- DB2 400
- SAP Hana

To create the source adapter:

1. On the **Setup** tab, under **Register**, select **Source Adapter**.
2. In **Source Adapter**, from the **Source Adapter** summary task bar, click **Add**.

A blank entry row is added to the Source Adapter summary section.

Complete the following steps in the Source Adapter details section.

3. In **Adapter Key**, enter a user defined identifier for the adapter.
4. In **Adapter Name**, enter a user defined name for the adapter.
5. In **Source System Type**, select the technology type:

Available options:

- Oracle
- MSSQL
- MYSQL
- Teradata
- DB2 UDB
- DB2 400
- SAP Hana

Based on the source system type, the following fields are populated automatically:

- ODI Package Name
 - ODI Project Code
6. In **Table Name**, specify the source table name.
 7. Click **Save**.

Adding Source Columns

To add a source column:

1. On the **Setup** tab, under **Register**, select **Source Adapter**.
2. Select the universal data adapter.

3. Click **Import Table Definition** to return all column details from the source table.
4. In **Context Code**, specify the context, and then click **OK**.

The ODI context code refers to the context defined in Oracle Data Integrator. A context groups the source and target connection information.

When the context is selected, all the columns of the source table are displayed in the Source Column tab.

5. **Optional:** in **Column Name**, enter the source column name.

By default, the Column Name is populated automatically when you select step 3 (Import Table Definition).

When using a view as a source for the Universal Data Adapter, don't include column names that are reserved words for the selected technology. For example, the word "Year" is a reserved word for Teradata, and should not be used in a view.

6. Based on the column type, from **Classification**, select the classification:

For example, select:

- Amount
- Year
- Period
- Period Number

7. **Optional:** in **Display Name**, enter the display name used for the import format.

By default, the Display Name is populated automatically when you select step 3 (Import Table Definition). Typically it is the function name of the column.

Defining Parameters

Use the Parameters tab to specify the list of parameters (filter) for the universal data adapter. The parameter definition includes a non-translated parameter name and a translated prompt. The prompt is used for display on the data rule page.

To add a parameter:

1. On the **Setup** tab, under **Register**, select **Source Adapter**.
2. Select the universal data adapter.
3. In the details section, select the **Parameters** tab.
4. From the **Parameters** task bar, select **Add**.

Entry fields are displayed for the Parameter Name, Parameter Data Type, Condition, Column Name, Default Value, and Parameter Prompt.

5. In **Parameter Name**, enter the parameter name.

For example, enter `p_actual_flag`. This is the name of the parameter in the ODI project.

6. In **Parameter Data Type**, select the data type of the parameter

Available data types:

- Char
- Number

- Date
7. In **Condition**, specify the type of the parameter:
 - **Explicit**—You are prompted to provide an explicit value in the Data Rule, which is matched exactly in the source table to pull data.
 - **Between**—You are prompted to provide a between values in the Data Rule, and these range of values are matched in the source table to pull data.
In—When this type is selected, you are prompted to provide multiple values in Data Rule, and these values are matched in the source table to pull data.
 - **Like**—You are prompted to provide a string in the Data Rule. Values starting with the string are matched in the source table to pull data.
 8. In **Column Name**, enter the column name to use as a filter.
 9. In **Default Value**, enter the value to default on the Data Load Rule screen.
When entering an "Explicit" value, use the 'X','XX','XXX' format.
When entering a "Between" value, use the 'X' and 'XX' format.
 10. In **Parameter Prompt**, enter the label to display for the parameter on the Data Rule screen.
When a new parameter is added and you are in a multi-language environment, then set the browser locale to the appropriate language and edit the prompts as needed in that language.
 11. Click **Save**.
 12. Click **Generate Template Package**.

This example shows a typical Structure of Fixed Asset Data Table:

Column Name	Data Type	Column Description
Asset_ID	Varchar	Asset Identification Code
Asset_Name	Varchar	Asset Name
Asset_Category	Varchar	Asset Category
Asset_Group	Varchar	Asset Group
Asset_PD	Date	Asset Purchase Date
Asset_ID	Date	Asset Inception Date
Asset_RD	Date	Asset Retirement Date
Asset_OC	Number	Asset Original Cost
Asset_AC	Number	Asset Additional Cost
Asset_RC	Number	Asset Retirement Value
Asset_AD	Number	Accumulated Depreciation
Asset_BV	Number	Asset Book Value
Asset_Location	Varchar	Asset Location

In the data structure of the fixed asset table, you could add the yellow marked columns as parameters to use as a filter in data rules.

On the Source Adapter screen using the table above, you might create parameters as follows:

The screenshot shows the 'Parameters' tab in the Source Adapter interface. It contains a table with the following data:

Parameter Name	Parameter Data Type	Condition	Column Name	Default Value
Asset_ID	Char			
Asset_Category	Char			
Asset_Group	Char			
Asset_Loc	Char			

Defining URLs for Drill Through

Oracle Hyperion Financial Data Quality Management, Enterprise Edition provides a framework for using URLs for drill through. You click the hyperlink in the amount cell, and the source system is launched in a new EPM Workspace tab or a new window. Multiple drill through URLs are provided to drill to different pages in the source system based on the import format mapping.

To add a drill through URL:

1. On the **Setup** tab, under **Register**, select **Source Adapter**.
2. In **Source Adapter**, select the **Drill URL** tab.
3. On the **Drill URL** task bar, select **Add**.

Entry fields are displayed for the Drill URL Name, Drill URL, Request Method, and Drill URL Prompt columns.

4. In **Drill URL Name**, enter a user defined name for the drill through URL.
5. In **Drill URL**, enter the **URL** used for the drill through.

Enter the URL without the server and port information. The URL must contain the parameter name and column name from the TDATASEG table enclosed in the symbol \$.

For example, enter: LEDGER_ID=\$ATTR1\$&GL_PERIOD=\$ATTR2\$.

In the above example the value of ATTR1 is passed as a value for the LEDGER_ID parameter, and ATTR2 is passed as the value for the GL_PERIOD parameter. Parameters are separated by the "&" character.

To specify the request-response between a client and server for the drill URL format, enter either:

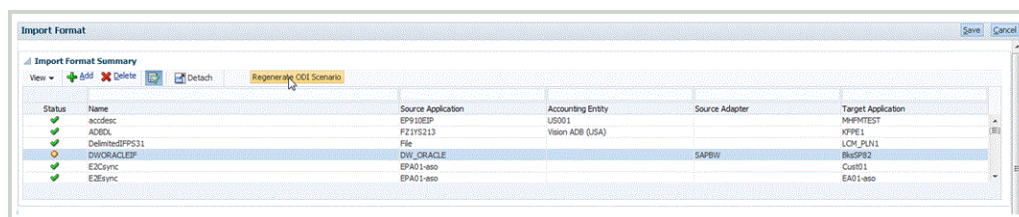
- GET—Form data is encoded into the URL. For example, specify: GET@http://www.oracle.com/. If no method is specified, then GET is the assumed request-response.
 - POST—Form data is displayed in the message body. For example, specify: POST@http://www.oracle.com/.
6. In the **Drill URL Prompt**, enter a user-defined prompt for the drill-through prompt. For example, enter Default.

Universal Data Adapter Source Systems and Import Formats

In the import format, you can map the source system and target application, and map source columns and target dimensions. After creating or making any changes to the import format, click "Regenerate ODI Scenario."

The screenshot shows the 'Import Format' dialog box. The top section, 'DWORACLEIF: Details', contains fields for Name (DWORACLEIF), Source System (ERP), Source Application (DW_ORACLE), Accounting Entity (a dropdown), Concatenation Character (a dropdown), and Drill URL (a text field with a search icon). The bottom section, 'DWORACLEIF: Mappings', shows a table with columns for Source, Target Dimension, and a list of mappings. The mappings are as follows:

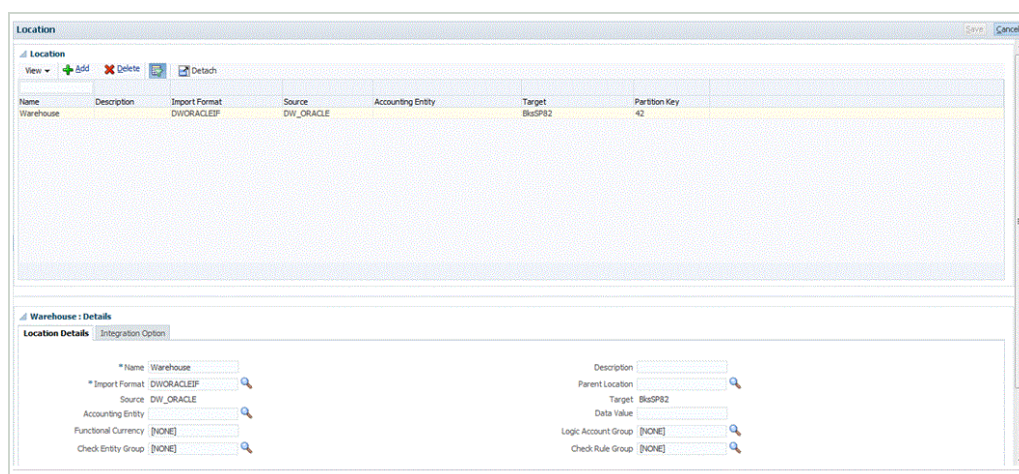
Source	Target Dimension
Account	Accounts
	Activities
	ACT_Activities
	ACT_Activities_intra
	ACT_CostCenters
	ACT_CostCenters_intra
	AllocationType
Amount	Amount
Company	CostCenters
	CO_Customers
	CO_Products
	Customers



Universal Data Adapter Source Systems and Locations

When working with universal data adapter source systems, no additional steps are required in Oracle Hyperion Financial Data Quality Management, Enterprise Edition locations.

The location typically supports the import format, which has been already created for the universal data adapter.



Data Load Mapping for Universal Data Adapter Related Sources

Data load mapping for the specified dimensions are similar to file or any other source system.

Data Load Rules for Universal Data Adapter Related Source

Data Load rules can be executed on an as needed basis to load balances from source to target applications.

Source Adapter Parameters

Parameters defined in the Source Adapter page are available for the user to select in data load mapping rules:

Source Adapter Details

Adapter Key: UDAORGL_1 * Adapter Name: UDAORGL_1
 * Source System Type: Oracle Data Source * Table Name: ORDERS
 ODI Package Name: UDAORGL_1 ODI Project Code: APF_ORACLE

Parameters

Parameter Name	Parameter Data Type	Condition	Column Name	Default Value	Parameter Prompt
City	Char	Explicit	City Name	New Delhi	Enter the City
Region	Char	Explicit	Region Name	North	Enter the Region

Category Mappings

You can create categories based on various target scenario dimension members.

To do this, you create required categories with assigned scenario members. These categories are captured during the Data Rule creation.

Category Mapping

Global Mapping

Category Key	Category	Description	Frequency	Target Category
1	Actual		Monthly	Actual
2	Budget		Monthly	Budget
3	Curr-QTR	QTR	Quarterly	Current
4	Current		Monthly	Current
5	Plan		Monthly	Plan
6	Plan_Act		Monthly	Actual

Period Mappings

On the Source Mapping tab, you select the source system for the universal data adapter, and specify the mappings as needed. In the following example, GL Year, GL Period and, GL Period Number are matched with the Source Columns classified as Year, Period Number, Period respectively.

Period Mapping

Source Mapping

* Source System: UDAORGL_1 Source System Type: Oracle Data Source Mapping Type: Explicit

Period Key	Period Name	Calendar	GL Year	GL Period	GL Period Number	Description
5/1/2015	May15	Accounting	2015	05	5	
9/1/2015	Sep15	Accounting	2015	09	9	

Note that the GL Year, GL Period, and GL Period Number columns have been mapped to the fields classified on the Source Adapter screen for the source table columns as shown below.

Column Name	Classification	Display Name
CUS_PERIOD_NUMBER	Period Number	Customer Data Period No.
CUS_PERIOD	Period	Customer Data Period
CUS_YEAR	Year	Customer Data Year
CUSTOMER_ID		CUSTOMER_ID
FREIGHT	Amount	FREIGHT

Multi-Periods

Oracle Hyperion Financial Data Quality Management, Enterprise Edition supports multi-period data loading for external table/view data.

To set up multiple period data loading for external table/view data:

1. On the **Source Adapter** screen, classify the source columns for **Year**, **Period**, and **Period Number**.

2. On the **Period Mapping** screen, define the source mappings.
For more information, see [Defining Period Mappings](#).

Period Key	Period Name	Calendar	GL Year	GL Period	GL Period Number	Description
5/15/2015	May 15	Accounting	2015	05	5	
9/15/2015	Sep 15	Accounting	2015	09	9	

3. On the **Data Load Rule**, from **Period Mapping Type**, select **Explicit**.
4. In **Calendar**, select the calendar specified on the Source Period Mapping screen.

The screenshot shows the 'Data Load Rule' configuration window. At the top, there's a 'Data Rule Summary' section with a table listing rules. Below this is a 'Details' section for a specific rule named 'udatest'. In the 'Details' section, there are several dropdown menus and checkboxes. A red arrow points to the 'Period Mapping Type' dropdown, which is set to 'Exploit'. Another red arrow points to the 'Include Adjustment Periods' dropdown, which is set to 'No'.

Status	Name	Category	Description
✓	udatest	Actual	

Details

Name: udatest
Category: Actual
Description:
Target Plan Type: Plan1
* Period Mapping Type: Exploit
* Calendar: Accounting
* Zero Balances: Exclude
* Include Adjustment Periods: No

Example 3-1 Executing the Data Load Rule

When executing the Data Load Rule, specify the range of periods.

The screenshot shows the 'Execute Rule' dialog box. It contains several options with checkboxes and dropdown menus. The 'Import from Source', 'Recalculate', 'Export to Target', and 'Execute Check' options are all checked. The '* Start Period' dropdown is set to 'Jan03' and the '* End Period' dropdown is set to 'Dec03'. The 'Export Mode' dropdown is set to 'Store Data'. At the bottom, there are 'Run' and 'Cancel' buttons.

Execute Rule

Import from Source ☒
Recalculate ☒
Export to Target ☒
Execute Check ☒
* Start Period Jan03
* End Period Dec03
Export Mode Store Data
Run Cancel

Integrating Data Relationship Management with FDMEE

Oracle Data Relationship Management functions as a hub where reporting structures are maintained, analyzed, and validated before moving throughout the enterprise. It is a change management solution for building and retaining consistency within master

data assets. Used in combination with Oracle Hyperion Financial Data Quality Management, Enterprise Edition, you can:

- export dimensions and hierarchies from an ERP system to Data Relationship Management.
- import data load mapping from Data Relationship Management for integration between ERP systems and EPM systems.

Setting up the Integration

Oracle Data Relationship Management integration with Oracle Hyperion Financial Data Quality Management, Enterprise Edition is enabled in the Target Application. When Data Relationship Management integration is enabled, you can export metadata to Data Relationship Management and import member mapping from Data Relationship Management.

To begin integrating Data Relationship Management with FDMEE, apply Data Relationship Management Release 11.1.2.4.340 Patch Number 23012151. Refer to the readme for patch 23012151 for install instructions. The FDMEE application template is installed as part of the Data Relationship Management installation. You use the template to configure the Data Relationship Management metadata to enable integration with FDMEE.

To register target applications:

1. On the **Setup** tab, under **Register**, select **Target Application**.
2. In **Target Application**, in the **Target Application** summary grid, click **Add**.
3. Add a target application or select a target application.
For information on adding a target, see [Registering Target Applications](#).
4. Click **Enable DRM**.
5. From **Application Details**, select the **DRM Options** tab.

Application Details
 * Name APP1 Type Custom Application Deployment Mode Not Applicable

Dimension Details Application Options **DRM Options**

View Refresh DRM Profiles Detach

Property Name	Value	Select
API Adapter URL	http://machine:5240/oracle/drm/apiadapter	
Web Service URL	http://machine:28080/oracle-epm-drm-webservices/Dr	
Username	css1	
Password	*****	
Import to DRM Profile (Metadata)	AR_CUSTOM_IP2	
Export from DRM Profile (Mapping)	AR_CUSTOM_Exp2	

6. In **Adapter API URL**, specify the full Data Relationship Management API adapter URL.
The adapter URL specifies the Data Relationship Management application to use for internal communication with the Web Service.
7. In **Web Service URL**, specify the URL of the full Data Relationship Management Web Service Definition Language (WSDL) of the web service used to access the Data Relationship Management web service.

Enter the machine computer name of the Web Services application, and the port number to which oracle-epm-drm-webservices is deployed.

8. In **Username**, specify the user name used to access the Data Relationship Management.
9. In **Password**, specify the password used to access the Data Relationship Management.
10. In **Import to DRM Profile (Metadata)**, specify the name of the import profile.

Import profiles update dimensions from Data Relationship Management, interface tables, and flat files. Import profiles also enable you to merge dimension updates merged with the shared library, or replace the entire contents of the dimension. Import profiles may map the columns from the source to their corresponding EPMA properties.

The Import profile contains the FDMEE target application name, IP address, and Data Relationship Management target application name.

The import profile can be overridden in the Integration option tab in locations.

Import profiles are built in Data Relationship Management. Data Relationship Management ships template profiles, which can be used as is or customized. Template profile names are stored in FDMEE.

11. In **Export from DRM Profile (Mapping)**, specify the name of the export profile.

An export profile enables you to export segment values and hierarchies from Data Relationship Management. Export profiles are based on export types which output data in different formats. Export types can include hierarchy, comparison, integration, version, property, and log exports.

The DRM Export Profile contains the FDMEE target application name, IP address, and Data Relationship Management target application name.

The export profile can be overridden in the Integration option tab in locations.

Export profiles are built in Data Relationship Management. Data Relationship Management ships template profiles, which can be used as is or customized. Template profile names are stored in FDMEE.

12. In **Target Application**, click **Save**.
13. Click **Refresh DRM Profiles** to show the list of Import Profiles, Export Profiles, and domains in list of values.
14. Define the dimension details.

See [Defining Application Dimension Details](#).

Data Relationship Management and FDMEE Locations

By default Oracle Data Relationship Management import and export profiles are defined at the application level. If you have multiple source systems from which to import and export Data Relationship Management data, you can select import or export profiles at the location level instead of the application level. This enables you to specify the Oracle Hyperion Financial Data Quality Management, Enterprise Edition location context so that only export mappings for the source values coming from a selected source system (for example, EBS or PeopleSoft) are made. Different Data Relationship Management import and export profiles can be selected on the Integration Options tab of the Location option.

Data Relationship Management and FDMEE Metadata Rules

Dimensions can be imported into 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 to other EPM applications. Dimensions map to hierarchies in Data Relationship Management using an Oracle Data Relationship Management import profile.

Oracle Hyperion Financial Data Quality Management, Enterprise Edition supports loading dimensions and hierarchies from ERP systems to Data Relationship Management. Key points of the metadata load include:

- You must specify the Data Relationship Management domain for each dimension. This domain is required if the Data Relationship Management integration is enabled for the target application. From the LOV, select the appropriate DRM Domain.
- The dimension name is the target dimension name.
FDMEE uses the effective date to determine the hierarchy version.
- FDMEE does not concatenate the version to the parent members because Data Relationship Management supports multiple hierarchies.

To create metadata rules for the Data Relationship Management:

1. On the **Workflow** tab, under **Metadata**, select **Metadata Rule**.



Note:

You cannot create multiple metadata rules for the same ledger or business unit for each target application.

2. From the **POV** bar, select the location to use for the metadata rule.
3. Click **Add**.
A blank line is displayed at the top of the Dimension Mappings summary grid.
4. In the Integration Option **Mapping** details area, from **Dimension**, select the dimension.
The dimensions listed are based on the import format.
When a Dimension is selected, the Dimension Classification field prefills.
5. In **DRM Domain**, select the domain name.
Domains are created by administrators and associated with versions by Data Manager role users or version owners.
Examples of domains include Accounts, Entities, Departments, Products, Employees, and Projects.
6. Define the mapping details for each dimension that you select.
7. Repeat steps 4-6 for each dimension.
8. Click **Save**.

Mapping Details

Dimension Entity (Product -Entity) Dimension Classification Entity

Prefix/Suffix Type Prefix/Suffix Value

Concatenate Name to Alias ☐

DRM Domain

Location **NK_DRM_PSFT** Period **Feb-33** Category **Actual** Source **PSFT APP1** Target **NKPLAN1**

Importing Data Load Mappings from Data Relationship Management

You can import member mappings from Oracle Data Relationship Management .Import member mappings from Data Relationship Management support merge or replace modes, along with validate or no validate options for target members.

Once the mappings have been validated and imported, the imported maps are displayed on the Data Load Mapping screen.

To import member mappings from Data Relationship Management:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Mapping**.
2. From **Import** options, select **Import from DRM**.
3. From **Import Mode**, select the import mode:
 - Merge—Overwrites the data in the application with the data in the data load file.
 - Replace—Clears values from dimensions in the data load file and replaces it with values in the existing file.

Select import mode and validation ✕

Import Mode: ☒ Merge ☐ Replace

Validation: ☒ Validate ☐ No Validate

Execution Mode: ☒ Online ☐ Offline

4. From **Validation**, select to validate the member mappings.
Validation ensures that all data in the imported GL has a corresponding mapping.
5. In **Execution Mode**, select the mode for executing the import:
 - Online—Process the import immediately.
 - Offline—Runs the import in the background.
6. Click **OK**.

The following message is displayed: "Import Map from DRM has been submitted successfully."

After the successful completion of the Import Process from DRM, data load mappings are available for edits and updates.

You can check the status of the import using Process Details.

Financial Close Management Integration with FDMEE

As part of the overall Financial Close process, Oracle Hyperion Financial Data Quality Management, Enterprise Edition enables data load from Enterprise Resource Planning (ERP) systems and files to EPM Applications. Oracle Hyperion Financial Close Management, which manages the period end close activities, initiates data load activities from within Financial Close Management. As the financial close processes are time-bound and work flow driven, Financial Close Management users trigger the processes based on task alerts and notifications.

Integration between FDMEE and Financial Close Management is achieved using the interactive mode. This type of integration enables Financial Close Management users to reach the Data Load Workbench of FDMEE by clicking a task link (URL) in Financial Close Management.

In this scenario, a user may respond to an email notification of a task to load data. When a user clicks the link (URL) in the mail, he or she can access the FDMEE from which to load data.

The Financial Close Management user transfers control to the Data Load Workbench and can continue with the rule execution process in an interactive way.

An example of the link (URL) is:

```
http://<YourServer>:19000/workspace/index.jsp?  
module=aif.launch&povLocationName=COMMA7DIM&povPeriodName=Jan-05&povCategoryName=A  
ctual&povRuleName=COMMA7DIM
```

Parameters passed to the Data Load Workbench are:

- povRuleName—Data Rule Name
- povLocation—Location
- povCategory—Category
- povPeriodName—Period

When the URL is defined in Financial Close Management, note the following:

- The URL requires all parameters.
- When a parameter has a space, enter the parameter with the spaces, but do not specify quotation marks ("()") around the parameter. (Parameters are passed in a standard URL request format).
- The delimiter is "&" in the URL.

Data Load, Synchronization and Write Back

Related Topics

- [Overview](#)
- [Synchronizing and Writing Back Data](#)

Overview

Oracle Hyperion Financial Data Quality Management, Enterprise Edition supports a variety of ways for importing data from a range of financial data sources, and then transforming and validating the data:

- **Data Loading**—define the mappings from a source system to a target system; drill through and view data in the ERP source system from an EPM target application; load data from file-based source systems to an EPM target application; define data load rules, which describe how to extract and push data from source to target systems.
- **Synchronizing**—move data between the EPM applications irrespective of the dimensionality of the application without having to create a data file from the EPM source application.
- **Write-back**—write-back budget data to the source system from all EPM applications to ERP applications. This feature offers significant advantages, such as writing back budgets created in Oracle Hyperion Planning to Peoplesoft or the Oracle E-Business Suite ERP GL, or moving adjustment journals from Oracle Hyperion Financial Management to ERP systems like E-Business Suite or Peoplesoft. Write-back budget data is also available to a file-based source system from Planning, Oracle ASO Essbase, and Essbase ESO applications.

Synchronizing and Writing Back Data

Data rules must be defined to load from an Enterprise Resource Planning (ERP) to an EPM application, synchronize data between two EPM applications, or write back from an EPM to an ERP system.

- **EPM Applications to EPM Applications (data synchronization)**—Moves data between EPM applications, for example, copying data from Oracle Hyperion Financial Management to Oracle Essbase for reporting.
- **EPM Applications to Enterprise Resource Planning (ERP) Applications (write-back)**—Moves data from EPM Applications to ERP applications, for example, writing back budgets created in Oracle Hyperion Planning to Peoplesoft or Oracle E-Business Suite ERP GL, or moving Adjustment Journals from Financial Management or the Accounts Reconciliation Manager to ERP Systems like E-Business Suite or Peoplesoft.

Data Synchronization

Data synchronization enables you to synchronize and map data between EPM source to target applications irrespective of the dimensionality of the application simply by selecting the source and target EPM application, and then mapping the data. Given the powerful mapping features already available, the data can be easily transformed from one application to another application.

Tasks enabled by the data synchronization:

- Create and modify synchronizations.
- Select source and target applications.
- Define mappings between sources and targets.

- Copy data from Oracle Hyperion Financial Management to Oracle Essbase for reporting purposes.
- Copy consolidated data from Financial Management to Oracle Hyperion Planning for future planning.
- Copy from one Financial Management application to another Financial Management used for different statutory reporting purposes.
- Write data from Oracle Enterprise Performance Management System to Enterprise Resource Planning (ERP) applications.
- Validate synchronizations.
- Execute synchronizations.
- View logs of synchronization activities.

At a high level, the steps to synchronize data in Oracle Hyperion Financial Data Quality Management, Enterprise Edition include:

1. **Import Format**—Select the source and target EPM System applications.
The dimensional mapping is also defined in the import format. You map the dimensions that comprise each source and target application so that the synchronization can recognize all relevant elements.
2. **Data Load Rule**—Define the source filter parameters.
3. **Execute**—When the data rule is executed, data from the source EPM System is extracted to a file. The data can be imported and processed using the data load workflow process.
4. **Export**—Synchronizes the data.

Using Data Load Rules for Synchronization

When synchronizing data, you specify which records (rows) are extracted from the source EPM Application to the target EPM Application.

Additionally, you can:

- Specify the source filter option to select a dimension, and then enter the filter criteria for each dimension.
- Browse and select members within a dimension.
- Load data to a single period or a range of periods.
- Add or change any target options for the application.

To define the data load details for a target EPM or Enterprise Resource Planning (ERP) system (file-based source system):

1. In **Name**, enter the data load rule name.
2. In **Category**, select a category.

The categories listed are those that you created in the Oracle Hyperion Financial Data Quality Management, Enterprise Edition setup, such as "Actual." See [Defining Category Mappings](#).

3. **Optional:** Enter a description.
4. In **Plan Type**, select the plan type.

FDME supports data loads for up to six plan types (including custom and Oracle Hyperion Planning applications.)

5. In **Period Mapping Type**, select **Default** or **Explicit**.
6. **Optional:** Add or change any source filter options.
See [Defining Source Filters](#).
7. **Optional:** Add or change any target options.
See [Registering Target Applications](#).
8. Click **Save**.

Using Default or Explicit Period Mapping Types

You can run data rules for one period or a range of them. The global or application period mappings of the target application are used as in the standard data load process.



Note:

When specifying a period range, make sure the start and ending periods are within a single fiscal year. When data ranges cross fiscal years, duplicate data results.

The source periods to be extracted are determined by the period mapping type.

Default Period Mapping

Default period mappings default to the list of source application periods using the application or global period mappings based on the period key. The list of source periods is added as Year and Period filters. For example, you can load data loading from Oracle Hyperion Financial Management to Oracle Essbase.

In the following example, Financial Management Application Period Mapping are loaded to Essbase Application Period Mapping for the period Jan-14 to Mar-15:

Table 3-17 Financial Management Application Period Mapping

Period	Year	Month
1/1/2014	2014	Jan
2/1/2014	2014	Feb
3/1/2014	2014	Mar

Table 3-18 Essbase Application Period Mapping

Period	Year	Month
1/1/2014	FY14	January
2/1/2014	FY14	February
3/1/2014	FY14	March

Using the example above, when the data is loaded, Oracle Hyperion Financial Data Quality Management, Enterprise Edition:

1. Uses the Essbase period mapping to determine the list of period keys: 1/1/2014, 2/1/2014, and 3/1/2014.
2. Determines the Financial Management period mapping, and inserts them into the AIF_PROCESS_PERIODS table.



Note:

You can have multiple source Financial Management period mappings to a given target Essbase period when the target application has a larger time frame (for example, Quarter) than the source period (for example, by Month).

3. Adds 2014 as a *Year* filter and Jan, Feb, Mar as *Period* filters.

Explicit Period Mapping

The Explicit method for loading data is used when the granularity of the source periods and target application periods are not the same.

For example, you need to load data from an Oracle Hyperion Financial Management application with monthly periods and an Oracle Hyperion Planning with quarterly periods.

In the following example, Financial Management Application Period Mapping are loaded to Oracle Essbase Application Period Mapping for the period Jan-14 to Mar-15:

Table 3-19 Financial Management Application Period Mapping

Period	Year	Month
1/1/2014	2014	Jan
2/ 1/2014	2014	Feb
3/1/2014	2014	Mar
4/1/2014	2014	April
5/1/2014	2014	May
6/1/2014	2014	June

Table 3-20 Planning Application Period Mapping

Period	Year	Month
1/1/2014	FY14	Q1
4/1/2014	FY14	Q2

The result of this method of mapping:

Table 3-21 Result of loading Financial Management Application Period Mapping to Planning Application Period Mapping

Period	Year	Month	Fiscal Year	Quarter
1/1/2014	2014	Jan	FY14	Q1
2/1/2014	2014	Feb	FY14	Q1
3/1/2014	2014	Mar	FY14	Q1
4/1/2014	2014	April	FY14	Q2
5/1/2014	2014	May	FY14	Q2
6/1/2014	2014	June	FY14	Q2

Using the example above, when the data is loaded, Oracle Hyperion Financial Data Quality Management, Enterprise Edition:

1. Determines the period key which is 1/1/2014, 4/1/2014.
2. Determines the Financial Management from the Financial Management period mapping to these period keys.
This mapping returns Year: 2014 and Periods: Jan. Feb, Mar, Apr, May, and Jun.
3. Adds Year and Period as source filters.

Defining Source Filters

Source filters enable you select a dimension, and then enter filter criteria for each dimension.

You can define source filter options to specify the subset of budget data to extract from your Oracle Hyperion Planning application and load it to the general ledger.

To define the EPM source filter options:

1. In **Source Filters**, click .
2. Select the **Dimension Name**.
3. To assign a filter condition, perform an action:
 - Enter a member name or filter condition in the Filter Condition text box.
For example, enter a member name or filter condition using Oracle Essbase syntax. Depending on the dimension, you can select one or more members as a filter condition used for extracting the budget data. For example, for the Entity dimension, you may select the following members: E1, E5, and E6. For information on Essbase syntax, see the *Oracle Essbase Database Administrator's Guide*.
 - Click  to display the Member Select screen and select a member using the member selector. Then, click **OK**.

The Member Selector dialog box is displayed. The member selector enables you to view and select members within a dimension. Expand and collapse members within a dimension using the [+] and [-].

The Selector dialog box has two panes—all members in the dimension on the left and selections on the right. The left pane, showing all members available in the

dimension, displays the member name and a short description, if available. The right pane, showing selections, displays the member name and the selection type.

You can use the V button above each pane to change the columns in the member selector.

You can also click **Refresh Members** to show the latest member list.

**Note:**

Assign filters for dimensions. If you do not assign filters, numbers from the summary members are also retrieved.

To use the member selector:

- a. In the list of available dimensions and members on the left, select a member and click .

- b. To deselect a member from the list of members, click .

- c. To add special options for the member, click  and select an option.

In the member options, “I” indicates inclusive. For example, “IChildren” adds all children for the member, including the selected member, and “IDescendants” adds all the descendants including the selected member. If you select “Children,” the selected member is not included and only its children are included.

The member is moved to the right and displays the option you selected in the Selection Type column. For example, “Descendants” displays in the Selection Type column.

**Tip:**

To clear all members from the list of selections, click .

- d. Click **OK** twice to continue defining the source filter details.

The selected member is displayed in Essbase syntax in the Filter Condition field.

4. Click **Save**.
5. **Optional:** Change or add any target options.

Defining Target Options

Target options define the options for importing a journal.



Note:

For information on the required target options for data load rules to write back, see [Defining Application Options for Essbase and Planning](#).

To define target Enterprise Resource Planning (ERP) source system options:

1. For Oracle Fusion and E-Business Suite source systems:
 - a. Select whether to create a budget journal.
 - b. Enter the **Budget** or click  to select the budget.
 - c. Enter the **Budget Organization** or click  to select the budget organization.
2. Click **Save**.
3. For PeopleSoft Enterprise Financial Management source systems, enter or click  to make selections:
 - a. **Budget Scenario**
 - b. **Ledger Group**
 - c. **Ledger**
4. Click **Save**.
5. Execute the data load rule.

Executing Data Synchronization

You execute the data synchronization by clicking **Execute** on the Data Load Rule screen. When the data load rule is executed, the data is extracted from the Oracle Enterprise Performance Management System application based on the filter criteria specified in the data rule. The data extracted is written to a data file. If multiple periods are specified in the source filter, then the amount columns are written as multiple columns. The data file contains the header record with the list of dimensions in the order in which they appear in the file. The file is created in the data folder with the name: *EPM App Name_PROCESS_ID.dat*.



Note:

When a data load rule is run for multiple periods, export step occurs only once for all of the period.

Data Import

The data import process imports the data file created during the extraction process. The import process evaluates the import format based on the header record in the file and mapping of the source to target dimension.

When the number and order of columns is determined, the column position is stored in the import format tables. File import expressions and scripts remain available during import.

Mapping and Export

During the mapping and export, Oracle Hyperion Financial Data Quality Management, Enterprise Edition:

- Imports data from Oracle Hyperion Planning and writes it to FDMEE staging table.
- Applies the necessary transformation in the staging table.
- Imports data from Planning and writes it to FDMEE staging table.
- Applies the necessary transformation in the staging table.
- Exports data from the FDMEE staging table to a general ledger interface table.

You use the mapping to format data in journal import tables.

- For Oracle Hyperion Financial Management, FDMEE extracts data and ICP transactions. FDMEE does not extract cell text or line-item detail.

After a successful execution of the data load rule used to write back data, log in to the general ledger source system and run the budget import process in the general ledger.

Drill-Through to Source

Oracle Hyperion Financial Data Quality Management, Enterprise Edition provides the framework to drill through from the Oracle Enterprise Performance Management System applications back to the general ledger from the EPM System source. Drill through works only for data loaded through FDMEE.

For example, you can drill through from where data was loaded from Oracle E-Business Suite to Oracle Hyperion Financial Management, and then from Financial Management to Oracle Essbase. When you viewing Essbase data in Oracle Smart View for Office, you can drill from the Essbase data cell and go to the FDMEE landing page.

Write-Back

Financial budgeting information often must be compared with and controlled with actuals and stored in the general ledger system. In Oracle Hyperion Financial Data Quality Management, Enterprise Edition, write-back functionality is available with the Export step of the data load process. In this way both loading to the Oracle Hyperion Planning application and write-back to General Ledger are performed in as a single consistent process.

Data Load Rules for Write-Back

You can write back budgets created in Oracle Hyperion Planning to their ERP General Ledger or a flat file. The write-back functionality is available with the Export step of the data load process. In this way both loading to the Planning application and write-back to General Ledger are performed in as a single consistent process.

For Oracle E-Business Suite, multiple ledgers can be used for the data load to write back with the following criteria:

- Ledgers must share the same chart of accounts.

- "Actuals" can be posted to multiple ledgers.
- "Budget" is associated with one ledger. One ledger can be posted from only one data load rule.

For Peoplesoft, "Actuals" can be posted without specifying the Budget Scenario. In addition, the write-backs can be made to multiple business units.

Available Source to Target Write-Backs

The target system for a write back must be a file-based application that uses dimensions that match the General Ledger Chart of Accounts. You can create the file-based application using the Custom Target Application feature (see [Creating a Custom Target Application](#)). When creating the application, ensure that the Export to File option is "Yes."

You can define data load rules to write back for Oracle Hyperion Planning, Oracle Essbase aggregate storage (ASO), and Essbase block storage (BSO) 11.1.2.x applications only as the source and for these target source systems:

- PeopleSoft Enterprise Financial Management
- Oracle E-Business Suite
- Fusion General Ledger
- Oracle Hyperion Financial Management to Oracle E-Business Suite

Budget & Actuals may be exported to Peoplesoft interface table: PS_HPYPB_ACCT_LN. Peoplesoft provides a Generate Journal Process only for Budget data. There is no process pre-defined to Generate Journal for Actual data. You must write a custom Journal template for the actual data.

Other Considerations:

- Data load to write back is supported only for Planning, Essbase aggregate storage, and Essbase block storage 11.1.2.x applications only. Applications created in earlier releases are not supported.
- For BSO Essbase applications, Oracle Hyperion Financial Data Quality Management, Enterprise Edition requires that the Period dimension be designated as "Dense." Other dimensions can be Dense, but Period must be designated as dense to write back.
- For E-Business Suite source systems, you can post to budgets with or without budget journals.
- Only monetary amounts may be written back to the source ERP's that provide a source adapter. If statistical amounts, or other data needs to be written back, then use scripting.
- You cannot create data load rules to write back to PeopleSoft Human Capital Management source systems.
- FDMEE loads data into the specific data interface table. You must then run the budget load routines provided by Oracle Fusion, E-Business Suite, or PeopleSoft Enterprise Financial Management.
- Allocation from a source amount to multiple target amounts is not provided.
- Data Load rules to write-back can be performed without first loading data from a General Ledger source to an EPM target.

- The category assigned to upgraded and write back rules is randomly assigned and plays no role in the function of the rule. To view an upgraded rule, select Show All Categories, which may reveal "hidden" rules.
- When specifying a period range, make sure the start and ending periods are within a single fiscal year. When data ranges cross fiscal years, duplicate data results.

This table shows available source-to-target write-back combination:

Table 3-22 Available Source to Target Write Backs

Source	E-Business Suite Budget	E-Business Suite Actual	PeopleSoft Budget	PeopleSoft Commitment Control	PeopleSoft Actual	SAP	JD Edwards
Planning	Out of Box	Out of Box	Out of Box	Out of Box	Custom	Use custom application.	Use custom application.
Essbase	Out of Box	Out of Box	Out of Box	Not supported	Custom	Use custom application.	Use custom application.
Financial Management	Out of Box	Out of Box	Out of Box	Not supported	Custom	Use custom application.	Use custom application.
Oracle Hyperion Profitability and Cost Management	Out of Box	Out of Box	Out of Box	Not supported	Custom	Use custom application.	Use custom application.
ARM	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported

Write-Back Security Access

The following interface tables require "write" security privileges for the data load rules to write back process:

E-Business Suite

- GL_INTERFACE
- GL_INTERFACE_CONTROL
- GL_BUDGET_INTERFACE

Standard PeopleSoft—PS_HPYPB_ACCT_LN

PeopleSoft Commitment Control

- PS_HYP_KK_BD_HDR
- PS_HYP_KK_BD_LN

See also [Source System Tables Used by FDMEE](#).

Creating Write-Back Mappings

To create write-back mappings:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. Enter the **Location Name** or click  to navigate and select the location.

3. Select the **Segment**.
4. Choose the type of mapping by selecting one of the following tabs: Explicit tab, Between tab, In Multi-Dimension, or Like tab.
 - **Explicit**—The source value is matched exactly and replaced with the target value. For example, the source value, “ABC” is replaced with the target value, “123.” “Explicit” write-back mappings are created the same for data load and data load to write back rules. See [Creating Mappings Using the Explicit Method](#).
 - **Between**—The range of source values is replaced with a single target value. For example, a range from 001 to 010 is replaced with 999. “Between” write-back mappings are created the same for data load and data load to write back rules. See [Creating Mappings Using the Between Method](#).
 - **In**—In mappings enable a list of non-sequential source values to be mapped to one target value. In this case, multiple values are mapped to one value within one rule, eliminating the need to create multiple rules (as is required for an Explicit map).
 - **Multi-Dimension**—Define member mapping based on multiple source column values.
 - **Like**—The string in the source value is matched and replaced with the target value. For example, source value Department is replaced by Cost Center A. See [Creating Mappings Using the Like Method](#).

Write-back mappings provide a means to remove or strip characters that were added during the data load process. “Like” write-back mappings are created like a reverse data load.

 Tip:

You can click **Refresh Values** to refresh the list of segment or chartfield values that appear in the drop-down list from the source system. This is especially helpful when creating “Explicit,” “Between,” “Like,” and “Multi-Dimension” mappings for data load rules to write back.

Defining Data Load Rules for Write-Back Scenarios (Data from EPM Cloud / Essbase Applications to Oracle Enterprise Resource Planning (ERP) Sources)

You create a data load rules to write back to extract budget data from application to a general ledger instance and ledger source.

You can create data load rules to write-back in these ways:

- Choose the Oracle Enterprise Performance Management Cloud application.
- Choose the Oracle Essbase aggregate storage (ASO) or Essbase block storage application (BSO).
- For Oracle Hyperion Public Sector Planning and Budgeting applications where you have consolidated personnel (HR) and non-personnel expenses in the aggregate storage cube, you pull information from the aggregate storage application. For nonpersonnel-related expenses you see only the total number (combination) in the aggregate storage application.



Note:

Public Sector Planning and Budgeting requires that you combine the regular planning results from an Essbase block storage cube, with the personnel position budget information in an Essbase aggregate storage cube to a new aggregate storage cube.

When performing the data load rule to write back for a Public Sector Planning and Budgeting application, you select the aggregate storage cube that you created in Public Sector Planning and Budgeting. You can also select an Oracle Hyperion Planning application as the source for the write-back.

The process at a high level:

1. Oracle Hyperion Financial Data Quality Management, Enterprise Edition imports data from EPM Cloud and writes it to a FDMEE staging table.
2. FDMEE applies the necessary transformation in the staging table.
3. Data is exported from the FDMEE staging table to a file-based application.
4. After a successful execution of the data load rule to write back, view the results of the balances transfer from the EPM Cloud application to the file using the Data Load Workbench.

To define data load rules to write back:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. Select the **Location Name** or click  to select the location name.
3. Click **Add** to add a data load to write-back rule.
4. Enter the **Name**.
5. **For EPM Cloud applications:** Select the **Plan Type**.
FDMEE supports data loads for up to six plan types (including custom and EPM Cloud applications.)
6. **Optional:** Enter a description.
7. Click **Save**.
8. Define the source and target options.

Defining the Source Filter Option

You can define source filter options to specify the subset of budget data to extract from your Oracle Hyperion Planning application and load it to the general ledger.

To define the source filter options:

1. On the **Workflow** tab, under **Data Load**, select **Data Load Rule**.
2. From the **Data Load Summary**, select the data load rule.
3. Select the **Source Options** tab.
4. In the **Source Filters** area, click .

5. Select the **Dimension Name**.
6. To assign a filter condition, perform an action:

- Enter a member name or filter condition in the Filter Condition text box. The selection syntax is based on the source system.

For example, if Oracle Essbase is the source system, enter the member name or filter condition using Essbase syntax. Depending on the dimension, you can select one or more members as a filter condition used for extracting the budget data. For example, for the Entity dimension, you may select the following members: E1, E5, and E6. For information on Essbase syntax, see the *Oracle Essbase Database Administrator's Guide*.

 Note:

For any dimensions not included in the source filter, Oracle Hyperion Financial Data Quality Management, Enterprise Edition includes level zero members. However, it is possible to have an alternate hierarchy in **For Planning applications** where a member that is a parent in the base hierarchy, is also a level 0 member in a shared hierarchy.

- Click  to select a member using the member selector, and then click **Browse**.

The Selector dialog box is displayed. The member selector enables you to view and select members within a dimension. Expand and collapse members within a dimension using the [+] and [-].

The Selector dialog box has two panes—all members in the dimension on the left and selections on the right. The left pane, showing all members available in the dimension, displays the member name and a short description, if available. The right pane, showing selections, displays the member name and the selection type.

You can use the Menu button above each pane to change the columns in the member selector.

 Note:

Assign filters for dimensions. If you do not assign filters, numbers from the summary members are also retrieved.

To use the member selector:

- a. In the list of dimensions and members on the left, select a member, and then click .
- b. To clear a member from the list of members, click .
- c. To add special options for the member, click , and then select an option.

In the member options, “I” indicates inclusive. For example, “IChildren” adds all children for the member, including the selected member. If you select “Children,” the selected member is not included, only its children are included.

The member is moved to the right and displays the option you selected in the Selection Type column. For example, “Descendants” displays in the Selection Type column.

 Tip:

To clear all members from the list of selections, click .

- d. Click **OK** twice to continue defining the source filter details.

The member that you selected is displayed in an Essbase syntax in the Filter Condition field.

7. Click **Save**.
8. Define the target options.

Defining Target Options (for Data Load Rules to Write back)

To write back to an Oracle E-Business Suite target, Oracle Hyperion Financial Data Quality Management, Enterprise Edition captures the Enterprise Resource Planning (ERP) options (such as "Budget," "Journal Source," and "Journal Category") on the Target Options tab. For example, if you are writing back to the E-Business Suite Actual, you must select the balance type "A," and then you also need to specify the journal source and journal category. Or if you are writing back to the E-Business Suite Budget journal, then select balance type "B," specify "Y" for the "Create Journal" option, and specify the journal source.

Required data rule E-Business Suite target options are shown in the table below.

Table 3-23 Required E-Business Suite Target Options

E-Business Suite Target	Balance Type	Budget Option
E-Business Suite Actual	A	Journal Source, Journal Category
E-Business Suite Budget Journal	B	Create Journal = Y, Journal Source, Journal Category, Budget
E-Business Suite Budget	B	Create Journal = Y, Budget

Required data load rule Peoplesoft target options are in the table below.

Table 3-24 Required Peoplesoft Target Options

PeopleSoft Target	Option
Peoplesoft Budget	Ledger Group, Ledger, Budget Scenario
Peoplesoft Actual	Ledger Group, Ledger

The target options define the options for importing a journal.

To define target Enterprise Resource Planning (ERP) source system options:

1. For Oracle Fusion and E-Business Suite source systems:
 - a. Select whether to create a budget journal.
 - b. Enter the **Budget** or click  to select the budget.
 - c. Enter the **Budget Organization** or click  to select the budget organization.
2. Click **Save**.
3. For PeopleSoft Enterprise Financial Management source systems, enter or click  to make selections:
 - a. **Budget Scenario**
 - b. **Ledger Group**
 - c. **Ledger**
4. Click **Save**.
5. Run the data load rule for the write back.

After you run the rule, perform required tasks in Fusion, E-Business Suite, and PeopleSoft Enterprise Financial Management.

Defining Financial Management to E-Business Suite Data Load Rules to Write-Back



Note:

For required Oracle E-Business Suite target options, see [Table 1](#).

To define Oracle Hyperion Financial Management to E-Business Suite data load rules to write back:

1. On the **Workflow** tab, under **Data Load Rule**, select **Data Load Rule**.
2. From the **Data Load Summary**, select the data load rule.
3. Select the **Target Options** tab.
4. In **Create Budget Journal**, select **Yes** or **No** to create the Budget Journal.
5. In **Budget**, select the budget value.
6. In **Journal Source**, select the source value.
7. In **Journal Category**, select the category value.
8. In **Budget Organization**, select the organization value.
9. In **Balance Type** select the **Actual** or **Budget** balance type.
10. Click **Save**.

Defining PeopleSoft Commitment Control Data Load Rules to Write-Back

To define PeopleSoft Commitment Control data load rules to write back:

1. Make sure that the PS_HYP_KK_BD_HDR and PS_HYP_KK_BD_LN interface tables have "write" privileges.
2. When writing back to PeopleSoft Commitment Control, in **Data Load Rule**, define a data load rule to write back by selecting parameters for the following:
 - Ledger Group
 - Ledger
 - Accounting Date—The accounting date indicates when a transaction is recognized, as opposed to the date the transaction actually occurred. The accounting date and transaction date can be the same. The accounting date determines the period in the General Ledger to which the transaction is to be posted. You can only select an accounting date that falls within an open period in the ledger to which you are posting. The accounting date for an item is normally the invoice date.
 - As of Date—The date on which data you are searching for or performing a function on is valid. For example, if you are defining a budget closing set, entering the "as of" date of 12/31/2016 instructs the PeopleSoft Commitment Control process to execute balance roll forward calculations based on the ledger amounts that are or were valid on that date.
 - Transaction Type—Select "check" to validate the rules or "post" to validate and post the rules.
 - Budget Transaction Type
 - Generate Parent Budget
 - Parent Budget Transaction Type
 - Default Entry Date—The maximum effective date budget definition that is less than the end date of the year is considered the correct budget definition to be used for all validations and entry event codes.
3. In **Data Load Rule**, define the source filters that identify the portions of budget data to load to PeopleSoft.
4. **Optional:** Delete previous data load rules to write back by selecting **Delete** on the **Execute Rules Options** screen.
5. Run the PeopleSoft Commitment Control data load rule to write back.

If you have selected a transaction type of "post," the rules are validated, the budget is written to Commitment Control interface tables and the posting process is initiated after writing the data. Any errors are forwarded to Public Sector Planning and Budgeting.

Running Data Load Rules to Write Back

After you create a data load rule to write back, you must run the data load rule to write back the data from the Planning application to the general ledger system.

To run a data load rule to write back:

1. From **Data Rule Summary**, select the data load rule.

2. Click **Execute**.

After the extraction process is successful, you must log in to Oracle General Ledger or PeopleSoft General Ledger and import the budget data.

3. To load data from the source EPM application, select **Import from Source**.

Select this option to review the information in a staging table, before exporting directly to the target general ledger system.

When you select "Import from Source," Oracle Hyperion Financial Data Quality Management, Enterprise Edition imports the data from the EPM target application, performs the necessary transformations, and exports the data to the FDMEE staging table.

4. To export data to the target general ledger system, select **Export to Target**.

FDMEE transfers the data into the general ledger system.

5. Click **Run**.

 Note:

After the rule runs successfully, view the information in the staging table. See [Staging Table Used for Import from Source](#).

6. After the rule runs, perform the required tasks in your general ledger system.

Exporting to Target

Use the Export to Target feature to export data to a target application, which is the Enterprise Resource Planning (ERP) application. Select this option after you have reviewed the data in the data grid and need to export it to the target application.

When exporting data for Oracle Hyperion Planning, the following options are available:

- **Store Data**—Inserts the value from the source or file into the target application, replacing any value that currently exists.
- **Replace Data**—Clears all data for the POV in the target, and then loads from the source or file. For example, when you have a year of data in your Planning application but are only loading a single month, this option clears the entire year before performing the load.
- **Add Data**—Adds the value from the source or file to the value in the target application. For example, when you have 100 in the source, and 200 in the target, then the result is 300.
- **Subtract Data**—Subtracts the value in the source or file from the value in the target application. For example, when you have 300 in the target, and 100 in the source, then the result is 200.

To submit the data load rule:

1. From the table action bar, in **Data Rule**, and choose the data load rule.
2. Click .
3. In **Execution Mode**, select the mode of exporting to the target.

Execution modes:

- online—ODI processes the data in sync mode (immediate processing).
- offline—ODI processes the data in async mode (runs in background).

Click  to navigate to the Process Detail page to monitor the ODI job progress.

4. In **Export**, select the export method.

Export options:

- Current Dimension
- All Dimensions
- Export to Excel

5. For **Current Dimension** and **All Dimensions** export methods, in **Select file location**, navigate to the file to export, and then click **OK**.

For the **Export to Excel** method, mappings are exported to a Microsoft Excel spreadsheet.

6. Click **OK**.

After you exported data to the target, the status of the export is shown in the Status field for the data load rule in the Data Load Summary.

Loading Data into Supported Source Systems

To complete the data load to write back process, you must perform required tasks in your supported source system.

Loading Data into Oracle E-Business Suite

If the target source system is Oracle E-Business Suite, you must import the budget data using Oracle General Ledger.

Refer to the Oracle Fusion and E-Business Suite user documentation for detailed information. At a high level, follow this process:

1. Launch Oracle General Ledger.
2. Run the Journal Import process.

If you select the Create Journal option:

- Data is extracted to the journal interface table (GL_INTERFACE).
- Oracle Hyperion Financial Data Quality Management, Enterprise Edition stores the process ID in the GROUP_ID column. You can check the data that was created by filtering on the process ID.
- If no errors occur, a journal entry is created.

If you clear the Create Journal option:

- Data is extracted to the budget interface table (GL_BUDGET_INTERFACE).
- Carefully select Budget Name (BUDGET_NAME) and Budget Organization (BUDGET_ENTITY_NAME) when loading data into the GL_BUDGET_INTERFACE table.
- If no errors occur, the budget balances are posted.

3. Review the journal or budget balances.

Loading Data into PeopleSoft Enterprise Financial Management

If the target is PeopleSoft, you must import the budget data using PeopleSoft General Ledger.

Refer to the PeopleSoft Enterprise Financial Management user documentation for detailed information. At a high level, follow this process:

1. Launch PeopleSoft General Ledger.
2. Run the Generate Journals process.
3. When generating a journal request, define these required options:
 - Accounting Definition Name—Select HYPDEFN
 - Template—Select HYP_STDBUD

The data is staged in the PS_HPYPB_ACCT_LN interface table. Oracle Hyperion Financial Data Quality Management, Enterprise Edition stores the process ID in the JRNL_LN_REF table. You can check the data that was created by filtering on the process ID.

4. Run the process.

If the process runs without errors, the journals are created.
5. Review the journal, create balancing entries, and then post the journal.

Loading Human Resources Data

This section explains how to load human resources data.

Understanding Human Resource Data Integration

Oracle Hyperion Financial Data Quality Management, Enterprise Edition supports loading human resource data from your PeopleSoft Human Capital Management source system into Oracle Hyperion Public Sector Planning and Budgeting applications.

You load human resource data by creating mappings in FDMEE to map the data from your PeopleSoft Human Capital Management source system to Public Sector Planning and Budgeting accounts.

FDMEE supports loading human resource data into Public Sector Planning and Budgeting applications. At this time, no other application types are supported.

If you have made revisions to properties in your Public Sector Planning and Budgeting application, loading human resource data from your source system into the application automatically overrides any modified application properties.

FDMEE supports loading data into the following Public Sector Planning and Budgeting models:

- Employee
- Position
- Employee and Position (both)

For a high-level process overview of the human resource integration, see [Extracting Human Resource Data](#).

Requirements

Before you begin your human resource integration, you must complete the following prerequisites:

- Run specific processes in PeopleSoft Human Capital Management. See [Running Processes in PeopleSoft Human Capital Management](#).
- Specify parameters in Oracle Hyperion Planning to enable data to be loaded into Oracle Essbase. See [Defining the Data Load Settings in Planning](#).

Running Processes in PeopleSoft Human Capital Management

To load future-dated salary data from PeopleSoft Human Capital Management, perform these steps in PeopleSoft Human Capital Management.

Refer to the PeopleSoft Human Capital Management user documentation for detailed information.

1. Run the Copy Job Data process (BPJBCOPY) to put a copy of data in PS_JOB, PS_COMPENSATION and PS_JOB_EARNS_DIST in PS_BP_JOB, PS_BP_COMPENSATION and PS_BP_JOB_ERN_DIST. Copies of these tables are used instead of human resource tables to ensure that the step increment additions to the data only occur in Oracle Hyperion Public Sector Planning and Budgeting tables and do not affect current human resource data when you run the step increment process.
2. Run the optional Load Step Increment (BPCMP107) process for budgets to generate future-dated job data in PS_BP_JOB and PS_BP_JOB_ERN_DIST.

Defining the Data Load Settings in Planning

In Oracle Hyperion Planning, you must define the data load settings before you can use Oracle Hyperion Financial Data Quality Management, Enterprise Edition to extract metadata and data from PeopleSoft Human Capital Management.

Data load settings enable you to choose the primary keys based of driver dimension members selected. You can specify other members that are not included in the selected driver dimension members. The data values of these members selected from the driver dimension as primary key are used to uniquely identify a row while finding the next available member from children of the selected member on the left hand side. The following figure shows the Data Load Settings window in Planning.

Data Load Settings	
Data Load Dimension	Budget Item
Driver Dimension	Account
Descendants(Human Capital Planning Accounts), Descendants(Segment I)	
Advanced Settings	
Data Load Dimension Parent	Driver Dimension Unique Identifiers
☐ Element Changes	Grade Step, Option Start Date
☐ FTE and Status Assignments	FTE Start Date, Assignment Start Date
☐ Allocation Assignments	Percentage Allocation, Allocation Start Date

See *Oracle Hyperion Planning Administrator's Guide* for detailed information.

Public Sector Planning and Budgeting Dimensions

The following Oracle Hyperion Public Sector Planning and Budgeting dimensions are populated by Oracle Hyperion Financial Data Quality Management, Enterprise Edition:

- Position
- Employee
- Element
- Job Code
- Entity

These dimensions are associated with the following tables:

Table 3-25 Dimensions populated by FDMEE

Dimension	Database Table
Position	AIF_PS_POSITION_DATA_STG
Employee	AIF_PS_JOB_STG
Job Code	AIF_PS_JOBCODE_TBL_STG
Element	AIF_PS_SAL_PLAN_TBL_STG AIF_PS_SAL_GRADE_TBL_STG AIF_PS_SAL_STEP_TBL_STG AIF_PS_EARNINGS_TBL_STG AIF_PS_BENEF_PLAN_TBL_STG
Entity	AIF_PS_DEPT_TBL_STG

For a complete list of Public Sector Planning and Budgeting tables, see [Staging Tables](#).

Smart Lists

Human Resource data such as salary information, union codes, and status are Smart Lists in Oracle Hyperion Public Sector Planning and Budgeting applications. Oracle Hyperion Financial Data Quality Management, Enterprise Edition automatically recognizes Smart Lists and populates the data accordingly.

FDMEE enables you to assign a prefix to Smart Lists in the Compensation Allocation point of view (POV). For general ledger integrations, you create metadata mappings and can optionally define member prefixes. For human resource integrations, you can optionally assign a Smart List prefix in the rule line mapping definition. You should ensure that member prefixes (used in a general ledger metadata mapping) are identical to Smart List prefixes (used in a human resource data rule mapping). For information on human resource rule mappings, see [Creating Mapping Definitions](#). For information on general ledger metadata mappings, see [Defining Metadata Rules](#).

For information about using Smart Lists in Oracle Hyperion Planning and Public Sector Planning and Budgeting, see the *Oracle Hyperion Planning Administrator's Guide* and the *Oracle Hyperion Public Sector Planning and Budgeting User's Guide*.

Using Loaded Data in Public Sector Planning and Budgeting

After loading human resource data, perform these tasks:

- For Classic Oracle Hyperion Planning applications, in Planning, refresh the application database as required. See the *Oracle Hyperion Planning Administrator's Guide*.
- For Planning applications administered in Enterprise Oracle Hyperion EPM Architect, navigate to the Application Library and redeploy the application. See the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*.
- To view the data and use it as part of Oracle Hyperion Public Sector Planning and Budgeting, see the *Oracle Hyperion Planning Administrator's Guide* and the *Oracle Hyperion Public Sector Planning and Budgeting User's Guide*.

Defining Human Resource Data Load Rules

At a high level, creating human resource data load rules that enable you to populate metadata and in Oracle Hyperion Public Sector Planning and Budgeting applications involves these steps:

- Create the human resource data rule and define the rule details. See [Defining the Rule Details](#).
- Create the rule line definitions and mappings manually or by running a script. See [Creating Rule Line Definitions by Using a Template Mapping](#) and [Creating Rule Line Definitions](#).
- [Creating Mapping Definitions](#).

Before you begin, ensure that you have performed all required tasks described in [Loading Human Resources Data](#).

- [Running Processes in PeopleSoft Human Capital Management](#).
- [Defining the Data Load Settings in Planning](#).

Before you create a human resource data load rule:

1. Register your human resource source system.
2. Register your target Public Sector Planning and Budgeting application.
3. Review the selected Source Accounting Entities.
4. Define import formats and locations.

For a high-level process overview, see [Extracting Human Resource Data](#).

Defining the Rule Details

To define rule details:

1. On the **Workflow** tab, under **HR Data Load**, select **HR Data Load**.
2. Enter or click  to select the **Location**.
3. In the Rule Name area, click **Add**.

4. In the Rule Definition area, enter the **Rule Name**.
5. Enter or click  to select the value of the **Scenario** dimension you identified when you registered the application. This enables you to classify and collect time-specific data.
6. **Optional:** Enter a **Description**.
7. Select the **Salary Assignment Option** to specify the salary assignment details to load by the database table.

Salary Assignment Options:

- **Standard**—Load data from PS_JOB table that is live on a given “As of Date.”
- **Auto Increment**—Load data from PS_BP_JOB table that is live on a given “As of Date” or is later than it.

 Note:

To use this option, perform these steps in [Running Processes in PeopleSoft Human Capital Management](#).

- **Standard - Include Future Dated**—Load data from PS_JOB table that is live on a given “As of Date” or is later than it.
8. Enter or click  to select the member of the **Version** dimension.
 9. Click **Save**.
 10. Next, create the rule line definition.

You have three options to create the rule line definition:

- Copy a rule—See [Creating Rule Line Definitions by Copying](#).
- Use a template to create the rule line definitions—See [Creating Rule Line Definitions by Using a Template Mapping](#).
- Create the rule line definitions manually—See [Creating Rule Line Definitions](#).

Creating Rule Line Definitions by Copying

You can create a rule line definition by copying a human resource data rule. Copying a rule copies all the POV rules and their mapping details.

 Note:

You cannot copy rules across applications.

To create rule line definitions by copying:

1. Perform the steps in [Defining the Rule Details](#).
2. In the Rule Definition area of the HR Data Load page, select **Create Mapping**, and then **Copy from another rule**.

3. Select the source rule.
4. Click **Copy**.
5. Click **OK**.
The POV rules and their mapping details are displayed in the Rule Line area.
6. Edit the mappings as necessary. See [Creating Mapping Definitions](#).

Creating Rule Line Definitions by Using a Template Mapping

You can use a template mapping to automatically create rule line definitions for Oracle Hyperion Public Sector Planning and Budgeting dimensions.

To create rule line definitions using a template:

1. Perform the steps in [Defining the Rule Details](#).
2. Select **Create Mapping**, and then select **Create Template Mapping**.
3. Select the **Language**.
4. Click **OK**.
5. Click **OK**.

The Rule Line Definition area is populated with POVs from the template.

6. Edit the rule line definitions or add new rule line definitions, as necessary. See [Editing Human Resource Data Load Rules](#).
7. Run the data load rule. See [Running Human Resource Data Load Rules](#).

Creating Rule Line Definitions

You create rule line definitions to map dimension members to an Oracle Hyperion Planning account.

To create rule line definitions:

1. In the Rule Line area, click .
2. In **POV Name**, enter a descriptive name for the point of view.
3. In **POV Type**, select the set of dimensions:
 - **Position Property**—Position dimension member properties
 - **Employee Property**—Employee dimension member properties
 - **Position and/or Employee Assignment**—Property and salary data associated with positions and employees.
 - **Compensation Allocation**—Allocation of salary costs to general ledger accounts for positions and employees.
 - **Jobcode Property**—Available if you created an Employee Only application in Planning, this dimension provides position and employee member properties.
 - **Salary Grade**—Rates for Element dimension members that correspond to salary grades.
 - **Salary Step**—Rates for Element dimension members that correspond to salary steps.

4. Define the mappings for the POV. See [Creating Mapping Definitions](#).
5. Repeat steps 1 - step 3 through for each POV you want to define.

Creating Mapping Definitions

You define mapping definitions to specify which Oracle Hyperion Public Sector Planning and Budgeting accounts to load with the extracted human resource data. For each dimension in the point of view, you must select a member, and then map the column in the source system to an Oracle Hyperion Planning account.

To create mapping definitions:

1. In the Rule Line area, click  to add a rule line mapping.
The Rule Line Mapping page is displayed.
2. In the Rule Line Point of View area, enter or click  to search for a member and map the members for dimensions in the point of view (POV).

You must enter a member for:

- **Budget Item**—For Budget Item dimensions, you may want Oracle Hyperion Financial Data Quality Management, Enterprise Edition to automatically create the budget item values. This option is available only for the Unspecified Budget Member. You must select a parent member if you plan to select the optional **Auto create flag** setting.

Note:

The parent member that you select must match what you selected in the Planning Data Load Settings window. See [Defining the Data Load Settings in Planning](#).

- **Year**
- **Period**

Selecting a member for all other dimensions is optional. If you do not select a member for a dimension, FDMEE loads the values as-is from the source system.

3. In the Rule Line Mapping area, define the source column to account mappings. To add a mapping, click .
4. Enter or click  to select the **Source Column**.
You select the source column in the PeopleSoft Human Capital Management table that contains the human resource data to load. For example, if you are creating a rule line mapping for Salary Grade, select the source column "Grade" in the PeopleSoft human resource table. The source columns are specific to the POV Type that you selected when you created the rule line definition.
5. Enter or click  to select the **Account**.

You select the Public Sector Planning and Budgeting account into which to load the extracted human resource data. For example, to load employee names and IDs, select Employee Name and Employee Number. For a list of all tables and columns, see [Staging Tables](#).

The Data Type and Smart List Name fields are automatically populated based on the Account you selected.

**Note:**

Several Smart Lists in Public Sector Planning and Budgeting applications are paired (for example, Account Segment and Account Segment Description). When you map from the source system to a paired Smart List, map one of the paired Smart List members (for example, Account Segment, but not Account Segment Description).

6. Click **Save**.
7. Click **Back** to return to the HR Data Load page.
8. Next, run the data load rule. See [Editing Human Resource Data Load Rules](#).

Managing Human Resource Data Load Rules

Perform the following tasks to manage your data rules:

- Edit data load rules—Typically, if you used a script to create the rule line definitions, you may want to add or edit the mappings. See [Editing Human Resource Data Load Rules](#).
- Run data load rules. See [Editing Human Resource Data Load Rules](#).
- Delete data load rules. See [Deleting Human Resource Data Load Rules](#).
- Check the data rule process details. See [Checking the Human Resource Data Load Rule Status](#).

Editing Human Resource Data Load Rules

To edit human resource data load rules:

1. On the **Workflow** tab, under **HR Data Load**, select **HR Data Load**.
2. Enter or click  to select the **Location**.
3. Select the rule to modify and change the rule details, rule line definition, and mapping settings. See [Editing Human Resource Data Load Rules](#).
4. Click **Save**.

Running Human Resource Data Load Rules

To extract and load data from PeopleSoft Human Capital Management, run the human resource data load rule that you defined.

When you run a human resource data load rule, you have several options.

- **Import from Source**—Oracle Hyperion Financial Data Quality Management, Enterprise Edition imports the data from the source system, performs the necessary transformations, and loads the data to the FDMEE staging table.

Select this option only in these situations:

- You are running a data load rule for the first time.
- Your data in the source system changed. For example, if you reviewed the data in the staging table after the export and it was necessary to modify data in the source system.

In many cases, your source system data may not change after you import the data from the source the first time. Therefore, you need not import the data.

- **Export to Target**—Exports the data to the target application.

Select this option after you have reviewed the data in the staging table and are sure you want to export it to the target application.

**Note:**

Select both options only in cases where you want to load the data directly into the target application.

To run human resource data load rules:

1. On the HR Data Load page, select the rule.
2. Click **Execute**.
3. Enter or select the **As of Date** from which to load data from the human resource source system.
4. To extract data from the source system, select **Import from Source**.
For information on staging tables, see [Staging Table Used for Import from Source](#).
5. To seed loaded data into your target Oracle Hyperion Public Sector Planning and Budgeting application, select **Export to Target**.
Selecting export to target exports the data to the target application.
6. If you selected **Export to Target**, perform these actions:
 - a. **To load data and/or metadata into a Classic Public Sector Planning and Budgeting application:** Select the **Load Type**:
 - Data—Loads only the data.
 - Metadata—Loads only the Public Sector Planning and Budgeting metadata.
 - Both—Loads the data and metadata.

**Note:**

Choose Metadata or Both as the load type on a new application; otherwise the data load fails.

- b. To load data and/or metadata into an Enterprise Oracle Hyperion EPM Architect Public Sector Planning and Budgeting application:
 - i. Select **Metadata** and run the data load rule.
 - ii. After the rule runs successfully, in Performance Management Architect, deploy the application.
 - iii. Return to FDME. On the HR Data Load page, locate the rule.
 - iv. Click **Execute** to run the data load rule, and then select **Data**.
- c. Select the **Department Load Option** to indicate whether to load all or specific departments and department data in the business unit:
 - **All**—Load data for all departments to the target application.
 - **Selected**—Load data for departments that you select to the target application. You can press the [Ctrl] key and select multiple departments.

 Note:

FDME merges data and does not "replace" balance data in a target application.

7. Click **Run**.

Data is loaded into your Public Sector Planning and Budgeting accounts.

Deleting Human Resource Data Load Rules

When you delete a human resource data rule, only the rule is removed. The extracted data or dimensions are not removed. You cannot delete a data rule if it is running.

To delete human resource data rules:

1. From the **Workflow** tab, under **HR Data Load**, select **HR Data Load**.
2. Enter or click  to select the **Location**.
3. Select the rule to remove, and then click **Delete**.
The Delete Confirmation dialog box is displayed.
4. Click **Yes**.
The rule is deleted.

Checking the Human Resource Data Load Rule Status

To check data load rule status:

1. On the **Workflow** tab, under **Monitor**, select **Process Details**.
2. From **Process Name**, select the HR load rule to check.



Note:

See the aif_<process_id>.log in the java.io.tmpdir folder (for example, c:\Oracle\Middleware\user_projects\epmsystem1\tmp\ai_f_2548.log) to view details of any rows rejected during the export step.

4

Logic Accounts

Related Topics

- [Overview of Logic Accounts](#)
- [Creating a Logic Group](#)
- [Creating Accounts In A Simple Logic Group](#)
- [Creating Complex Logic Accounts](#)

Overview of Logic Accounts

Logic accounts are dynamically generated accounts, and are used to calculate supplemental values that are not provided in source files. Logic groups are associated with a target application. (The logic group list of values is filtered in the location user interface based on the target application in which it was created.) Like all other source accounts, logic accounts can be mapped and loaded into target systems. Logic accounts are used for various functions:

- **Statistical loading**—Map one source account to multiple target accounts
- **Conditional mapping**—Map a source account based on its value
- **Arithmetic mapping**—Perform arithmetic operations on source values

Creating a Logic Group

The first step in creating logic accounts is to create a logic group. The logic group is then assigned to one or more locations. When a source file or source system is loaded to a location, logic accounts are generated when the logic group is assigned to a location.

A logic group must be defined as simple or complex. A simple logic group enables you to derive logic items only from the source account dimension. A complex logic group enables you to derive logic items from any combination of dimensions.

To create a logic group:

1. On the **Setup** tab, under **Data Load Setup**, select **Logic Group**.
2. In **Target Applications**, select the target application for this logic group.
3. In **Logic Group**, select **Add**.
A row is added to the grid.
4. In **Logic Group Name**, enter a unique name.
Optional: In **Description**, enter a description of the logic group.
5. In **Logic Type**, select **Simple Logic** or **Complex Logic**.
6. Click **Save**.

Creating Accounts In A Simple Logic Group

Within a simple logic group, you can create individual logic accounts.

To create accounts in a simple logic group:

1. On the **Setup** tab, under **Data Load Setup**, select **Logic Group**.
2. From the **Logic Group** summary grid, select the logic group.
The logic accounts currently contained in with the selected logic group are listed.
3. From the **Logic Items** grid, click **Add**.
4. Provide the requested information.

See [Logic Group Fields](#).

Logic Group Fields

Logic accounts consist of the following fields:

- Item
- Description
- Criteria Type
- Criteria Value
- Operator
- Value/Expression
- Seq
- Export

Item

Specify the name of the logic account using the item field. The logic account that is named in the item field is displayed in the Workbench grid as the source account. This same account can be used as a source in a mapping rule. Oracle recommends that you prefix the names of logic accounts with an "L" or some other character to indicate that an account came from a source file, or was generated from a logic rule. Logic accounts can only be loaded to a target application when they are mapped to a target account.

Description

The description that you enter in the Description field is displayed in the Account Description field in the Workbench.

Criteria Type and Criteria Value

The operator in the Criteria Type field works with the source account that is specified in the Criteria Value field to determine from which source accounts the logic account is derived. One logic account can be derived from multiple source accounts.

Valid values for the Type field:

- Between

- Like
- In

Specifying Source Accounts

Between (Criteria Type)—Used when a range of source accounts is specified in the Criteria Value field. Separate the accounts that specify the range by a comma.

Table 4-1 Between Type field and example of the corresponding Criteria Value Field values.

Type Field	Criteria Value Field
Between	1000,1999

Like (Criteria Type)—Used when the source accounts in the Criteria Value field contain wildcard characters. Use question marks (?) as placeholders and asterisks (*) to signify indeterminate numbers of characters.

Table 4-2 Like Type and Examples

Type Field	Criteria Value Field
Like	1??0
Like	10*

In (Criteria Type)—Used to include one source account or a list of non-sequential source accounts.

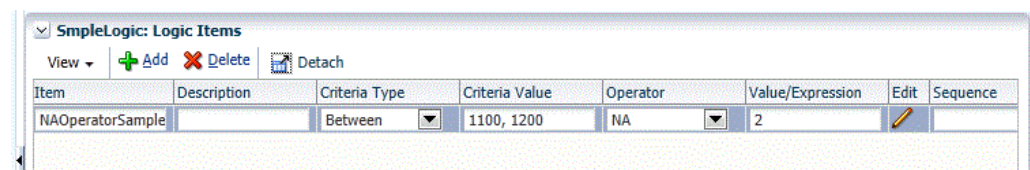
Table 4-3 In Type and Examples

Type Field	Criteria Value Field
In	1000
In	1000,1005,2001

Operator and Value/Expression

NA Operator

NA (no operator)—If NA is specified, the source accounts that are specified in the Criteria Value field are summed. For example when the Type = Between and the Criteria Value = "1100,1200," then Oracle Hyperion Financial Data Quality Management, Enterprise Edition creates one new account summarizing the values for source accounts that are between 1100 and 1200 for each entity where the accounts exist in the source file.

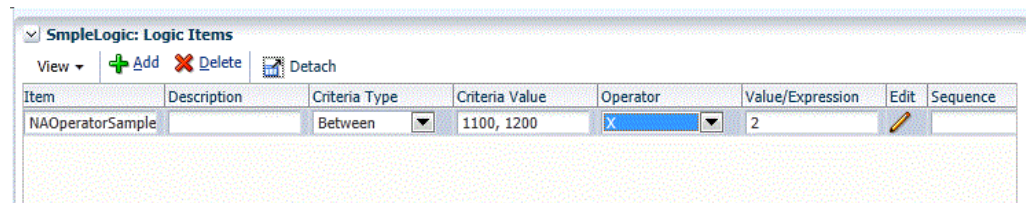


Math Operator

Math Operators (+, -, x, /)—If a math operator is selected, then the new logic records has an amount that equals the original amount is calculated with the specified Value/Expression. For example, when the operator “x” was selected and 2 is entered in the Value/Expression field, then the new record has an amount two times the original amount.

Use a numeric operator to perform simple mathematical calculations:

- NA (no operator)
- + (addition)
- - (subtraction)
- X (multiplication)
- / (division)
- Exp (expression operators)
- Function—see [Function](#)



In this example, one logic account is created because one Entity had a row meeting the account criteria.

Expressions and Functions

An expression enables you to perform the following in the Value/Expression field:

- Execute a simple math equation.
- Use a CURVAL parameter to specify the value of a logic account operation. The CURVAL parameter can be used within expressions, as it can within logic functions, except that, with expressions, CURVAL must be enclosed in pipes. For example, the CURVAL expression includes the specification of an account using the notation of |Account| and the specification of a value using POV details of entity, category, period and account.

Functions enable the use of simple logic with if/else using Jython syntax. Jython commands can be used in a function and can be more complex than an expression.

Exp

Use Expression operators to execute custom logic expressions, which are defined in the Value/Expression field. Logic expressions, which cannot use variables or If statements, are simpler than logic functions. Except for |CURVAL|, expressions do not have built-in parameters. For expressions, you do not need to assign a value to RESULT.

Expressions execute faster than logic functions. You can use the Oracle Hyperion Financial Data Quality Management, Enterprise Edition Lookup function within expressions, as it is used within logic functions. To write a custom expression, double-click the Value/Exp field to open the expression editor.

```
|CURVAL| + |810| + |238|
```

The function above uses the FDMEE Lookup function to add two source accounts to the value of the logic account. Notice that the CURVAL parameter can be used within expressions, as it can within logic functions, except that, with expressions, CURVAL must be enclosed in pipes.

```
(|CURVAL| + |000,10,09/30/01,810|) * 100
```

The function above uses the FDMEE Lookup function to add a source account (810) and a source account from a specified center, FDMEE category, and FDMEE period to the value of the logic account, and then multiplies the resulting sum by 100.

Function

Use function operators to execute a custom logic function defined in the Value/Expression field.

To write a function, select Function from the Operator drop-down list in the Logic Item line, and then click the edit icon to open the edit window. Logic functions are usually used for conditional mapping and other complex operations that involve multiple source accounts. Logic functions enable the use of Jython commands including variables, if/elif/else statements, numeric functions, and other Jython constructs.

The logic function enables the use of predefined function parameters, and also requires that you assign a value to the RESULT variable so that a value can be updated for the newly created logic account. The following function parameters can be used in a logic function, and these do not require using the “|” notation:

Table 4-4 Function Operators and descriptions

Function Operator	Description
CURVAL	Source value of the logic account operation
StrLocation	Active location name
StrCenter	Logic account entity
StrCatKey	Active category key, not the name. You need too lookup the category key in the database to use this parameter.
StrPerKey	Active period
Entity, Category, Period, Account	Lookup notation may be used in a logic function. This is the same notation provided in the logic expression.
Skip	If “Skip” is assigned to the keyword RESULT, then the logic account is not created.

You can define function parameters in uppercase, lowercase, or mixed case letters. However, the keyword RESULT must be in uppercase letters.

Assigning Function Return Values

The result of a Logic Function must be assigned to the keyword **RESULT**. If a return value is not assigned to the **RESULT** keyword, then the logic engine automatically sets the value of result to zero. This causes the calculation to be skipped and the logic account is not created.

The following function assigns the result of the logic account calculation (using the **CURVAL** parameter) to the logic account (**RESULT**) when the logic account calculation returns a value greater than zero. If the first condition is not met, then the logic account is not created because of the keyword "Skip."

```
if CURVAL > 0:  
  
    RESULT = CURVAL  
  
else:  
  
    RESULT = "Skip"
```



Note:

You must use the Jython notation and indentation for the logic function.

The following function only assigns the result of the logic account calculation to the logic account when "10" is the active **FDME**E category key.

```
if StrCatKey == "10":  
  
    RESULT = CURVAL  
  
else:  
  
    RESULT="Skip"
```

This function assigns the result of the logic account calculation to the logic account only when the **Criteria Account Entity** is "000."

```
if StrCenter == "000":  
  
    RESULT = CURVAL * 100  
  
else:  
  
    RESULT="Skip"
```

This function uses the **FDME**E Lookup function to add a source account (810) to the value of the logic account if the current **FDME**E period is "Dec 2013."

```
if StrPerKey == "12/31/2013":  
  
    RESULT = CURVAL + |810|  
  
else:  
  
    RESULT="Skip"
```


This function uses the FDMEE Lookup function to add another source account from a different Entity, FDMEE category, and FDMEE period to the value of the logic account when the active location is "Texas."

```
If StrLocation == "Texas":
```

```
    RESULT = CURVAL + |000,10,09/30/13,810|
```

```
else:
```

```
    RESULT="Skip"
```

Value/Expression

To perform calculations and thereby, to derive values for a logic account, you select an operator, from the Operator field, to work with the Value/Expression value.

Seq

This field specifies the order in which the logic accounts are processed. Order specification enables one logic account to be used by another logic account, provided that the dependant account is processed first.

Export

A Yes-No switch determines whether a logic account is considered an export account and therefore is subjected to the conversion table validation process. If the switch is set to Yes, then you must map the logic account.

Creating Summarized Logic Accounts

By default, a logic account is created for each center in the trial balance. For example, when the Criteria Value field is 12300, the result is a logic account created for each source center that is associated with account 12300.

You can create a logic account that summarizes multiple source centers by, in the Criteria Value field, placing a semicolon after the account name and entering the number that identifies the number of characters to group by.

For example, when the value in the Criteria Value field is 12300;4, the result is a summarized account that includes all source centers that have the same first four characters in the source center name. The source center assigned to account 12300 is the four characters that start at position 1. In addition, when the value in the Criteria Value field is 12300;3;4, the result is a summarized account that includes all source centers that have the same three characters in the source center, starting at position 4. The source center assigned to account 12300 is the three characters that start at position 4.

To create a logic account that summarizes all source centers by, in the Criteria Value field, place a semicolon after the account name and enter a text value. This hard-coded text value becomes the center for the summarized logic account. For example, when the value in the Criteria Value field is 12300;Dept100, then the result is a summarized account that includes all source centers. The source center assigned to account 12300 is Dept100.

Creating Complex Logic Accounts

Individual logic items are defined within a complex logic group. Each of the fields for a complex logic rule operates the same as a simple logic rule except for the Criteria Value and Include Calc fields. Complex logic accounts enable the user to enter a criteria value that includes other dimensions besides the account dimension. In addition, you can specify a “group by” and “group level” to alter how the generated logic accounts are displayed within the Workbench.

Criteria Value

To enter criteria for each dimension, click the Criteria Value icon to open the criteria form. The logic item is created only from the source line items that meet the specified criteria for each dimension. Descriptions of each complex logic criteria field is as follows:

Dimension

This field enables the selection of any enabled source dimension. You can select each dimension only once.

Criteria Type

This field works in conjunction with the Source Dimension and Criteria Value fields to determine from which source values the logic items are derived. Criteria types available are In, Between, and Like. The Criteria Type determines how the criteria value is interpreted.

Criteria Value

The criteria type uses this field to determine the members to include in the logic calculation for any given logic dimension.

Group By

When viewing the derived logic item in the Workbench, the Group By field enables the logic item to override the displayed member in the appropriate dimensions field. You can override to group the dimension based on the value entered in the Group By field. Use this field to hard code the returned member, or append hard-coded values to the original members by entering a hard-coded member and an asterisk (*) in the Group By field.

For example, by placing the word “Cash” in the row with account selected for dimension, the Import form displays “Cash” in the Account field for the logic item. If you place “L-*” in the Group By field, the import form displays “L-1100” where 1100 is the original account that passed the logic criteria.

If you enter no value in the Group By field, no grouping occurs for this dimension, and a separate logic item is created for each unique dimension member.

Group Level

When viewing the logic item in the Workbench, the Group Level field works with the Group By field to override the displayed member in the appropriate dimensions field. This field accepts only numeric values.

When you enter a value of 3 in the Group Level field, the left three characters of the Group By field are returned. If no value is entered in the Group By field, then when you specify 3 in the Group Level field, first three characters of the original source dimension member are returned. The logic items displayed on the Import form can be grouped to the desired level.

For example, when you enter L-* in the Group By field, the logic item displays in the Import form as "L-1100," where 1100 is the original account that passed. When viewing the logic item in the Workbench, the Group Level field works with the Group By field to override the displayed member in the appropriate dimensions field. This field accepts only numeric values.

+ displays "L-11." If you enter the Group level1 for this row, then the Import form displays "L-1."

Include Calc

If it meets the logic item criteria, the Include Calc field enables the logic item to include previously calculated Oracle Hyperion Financial Data Quality Management, Enterprise Edition values in its calculations.

Note:

Each logic item has a sequence attached, and the logic items are calculated in this sequence. If the second, or later, logic item has this field enabled, then any previously calculated logic items are included, provided they meet the logic criteria.

Complex Logic Example 1: CashTx

Table 4-5 Complex Logic Example 1: CashTx

Dimension	Criteria Type	Criteria Value	Group By	Group Level
Account	Like	11*	Cash	0
Entity	Like	Tx	Texas	0
ICP	Between	00,99	ICP	0
UDI	In	00,01,02	UD1	0

The first row specifies that any accounts that begin with "11" are included in the calculated result for "Calc Item: CashTx."

The second row further qualifies the results by specifying that the source record must also have the entity like "TX."

The third line reduces the results to only those source records that have an ICP value between 00 and 09.

The last line reduces the results to only those source records that have a Custom 1 (UD1) of either: 00, 01 or 02. Imported lines that do not meet the listed criteria are excluded from the calculated results.

In the following table, only one new logic item is derived from multiple source records. Using the preceding graphic example as the logic criteria, and the first grid that follows as the source line items, you can see how Oracle Hyperion Financial Data Quality Management, Enterprise Edition derives the value of a single logic item. Note the Group By field. Each Group By field includes a hard-coded value. Therefore, for every line that passes the specified criteria, the original imported member is replaced with the member listed in the Group By field.

Sample Imported Values

Table 4-6 Imported Values Sample

Account	Entity	ICP	UD1	Amount	Include or Exclude
1150	Tx	07	01	50,401.07	Include
1176	Tx	04	02	10,996.00	Include
1201	Tx	01	00	500.00	Exclude

Sample Imported Account Names

Table 4-7 Sample Imported Account Names

Account	Entity	ICP	UD1	Amount
Cash	Texas	ICP	UD1	50,401.07
Cash	Texas	ICP	UD1	10,996.00

Oracle Hyperion Financial Data Quality Management, Enterprise Edition groups and summarizes the rows that include identical member combinations and thus creates the following result:

Final Result

Table 4-8 Imported Account Names and Numbers

Account	Entity	ICP	UD1	Amount
Cash	Texas	ICP	UD1	61,397.07

Complex Logic Example 2: CashTx

Table 4-9 Complex Logic Example 2

Dimension	Criteria Type	Criteria Value	Group By	Group Level
Account	Like	11*	Cash	0
Entity	Like	Tx	Texas	0
ICP	Between	000,100	*	2
UDI	In	00,01,02	UD1-*	0

The first row in the preceding table specifies accounts that begin with “11” are to be included in the calculated result for “Calc Item: CashTx.”

The second row further qualifies the results by specifying that the source record must also have the entity like “TX.”

The third line reduces the results to only those source records that have an ICP value between 000 and 100.

The last line reduces the results to only those source records that have a Custom 1 (UD1) of either: “00,” “01.” or “02.” Any imported line that does not meet all of the listed criteria is excluded from the calculated results.

In the following tables, two logic items are derived from the source records because of the values entered in the Group By and Group Level fields. Two of the Group By fields have hard-coded values listed and two have an asterisk. Therefore, for every line that passes the specified criteria, the original imported members for the Account and Entity dimensions are replaced with the member listed in the Group By field. The other dimensions return all or part of the original members based on the Group Level entered.

Sample Imported Values

Table 4-10 Sample Imported Account Numbers

Account	Entity	ICP	UD1	Amount	Include or Exclude
1150	Tx	070	01	50,401.07	Include
1176	Tx	040	02	10,996.00	Include
1121	Tx	045	02	9,050.41	Include
1201	Tx	100	00	500.00	Exclude

Logic Members

Table 4-11 Logic Members Imported Account Names

Account	Entity	ICP	UD1	Amount
Cash	Texas	07	UD1-01	50,401.07
Cash	Texas	04	UD1-02	10,996.00
Cash	Texas	04	UD1-02	9,050.41

Oracle Hyperion Financial Data Quality Management, Enterprise Edition groups and summarizes the rows that include identical member combinations and thus creates the following result.

Final Result

Table 4-12 Final Result of Imported Account Names and Numbers

Account	Entity	ICP	UD1	Amount
Cash	Texas	07	UD1-01	50,401.07
Cash	Texas	04	UD1-02	20,046.41

5

Check Rules

Related Topics

- [Overview of Check Rules](#)
- [Creating Check Rule Groups](#)
- [Creating a Check Rule](#)
- [Creating Check Entity Groups](#)

Overview of Check Rules

System administrators use check rules to enforce data integrity. A set of check rules is created within a check rule group, and the check rule group is assigned to a location. Then, after data is loaded to the target system, a check report is generated.

If a check entities group is assigned to the location, then check report runs for all entities that are defined in the group. If no check entities group is assigned to the location, the check report runs for each entity that was loaded to the target system. Oracle Hyperion Financial Data Quality Management, Enterprise Edition check reports retrieve values from the target system, FDMEE source data, or FDMEE converted data.

FDMEE analyzes the check report and inserts a status entry in the process monitoring table. The location associated with the report shows a status of True only when all rules within the check report pass. For rules used only for warning, no rule logic is assigned.

Check reports run as data is loaded. You can also run the reports manually.

Note:

Check rules are not applicable when loading to Accounts Reconciliation Manager.

Note:

If the Entity dimension has shared hierarchies, then members needs must specified in parent.child format in the check entity group or data load mappings for check rules to work with Oracle Financial Consolidation and Close Cloud and Oracle Tax Reporting Cloud.

Creating Check Rule Groups

To create check rule groups:

1. On the **Setup** tab, under **Data Load Setup**, select **Check Rule Group**.
2. In the **Check Rule Group** summary grid, click **Add**.
A row is added to the top grid.
3. In **Check Rule Group Details**, enter a name for the group in the **Name** field.
Optional: In **Description**, enter a description of the group.
4. Click **Save**.

Creating a Check Rule

Each line of a Check Rule report represents a check rule.

To create check rules:

1. On the **Setup** tab, under **Data Load Setup**, select **Check Rule Group**.
2. Optional: In **Check Rules**, select the POV Location, POV Period, or POV Category.
See [Using the POV Bar](#).
3. In the **Check Rule Group** summary grid, select the check rule group.
4. In the **Rule Item** details grid, click **Add**.
A row is added to the grid.
5. In each field, enter check rule information:
 - **Display Value**—See [Display Value](#).
 - **Description** (optional)—See [Description](#).
 - **Rule Name**—See [Rule Name](#).
 - **Rule Text**—See [Rule Text](#).
 - **Category**—See [Category](#).
 - **Sequence**—See [Sequence](#).
 - **Rule Logic** (optional)
6. Click **Save**.

Example 5-1 Display Value

The Display Value field, which controls how Oracle Hyperion Financial Data Quality Management, Enterprise Edition formats the data rows of check reports, is used to select target accounts or report format codes. For fields that contain report format codes, no value lookup is attempted.

Example 5-2 Browse for Target Account

This option, which displays the **Search and Select: Target Value** screen, enables you to search and insert a target account (from a list of target-system application accounts) into the check rules form.

Example 5-3 Select Format Code

This option enables you to enter format codes into the Target Account column.

Format codes determine the display of check reports.

Table 5-1 Format Codes and Corresponding Actions Performed on Check Reports

Format Code	Action Performed on Check Reports
#ModeList	Sets the report to display the Display Value, Description, and Amount column values. The system defaults to #ModeRule if nothing is specified.
#ModeRule	(Default) Sets the report to display the Rule Name, Rule Text, and Amount column values. The report evaluates each expression of the Rule Logic column and tests the True or False condition of each rule. The status of each rule (OK or Error) is displayed in the report.
#Title	Inserts the text of the associated Description field as a title line on the check report.
#Subtitle	Inserts the text of the associated Description field as a subtitle line on the check report.

Example 5-4 Description

Displayed only for check reports in #ModeList mode, the Description column displays account descriptions (which may be designated as titles or subtitles).

Example—Description

Out-of-Balance Account

Example 5-5 Rule Name

Displayed only for check reports in #ModeRule mode, the Rule Name column, stores identifiers for check rules. Rule Name values should be unique and easy to identify.

Example—Rule Name

Out-of-Balance Check

Example 5-6 Rule Text

Displayed only for reports in #ModeRule mode, the Rule Text column defines the logic behind rules. In check reports, the primary statement for a rule is the text of the Rule Text field associated with the rule.

Example—Rule Text

This account must be between [+10 and -10].

Example 5-7 Category

In the Category column, select the FDMEE category to restrict a check rule to one FDMEE category. The rule is displayed in the check report only when the FDMEE

category that is selected in the Category field associated with the rule is the FDMEE category set in the POV. To display the check rule in the check report regardless of the category set in the POV, you must select **All**.

Example 5-8 Sequence

Sequence column values (numbers) determine the order in which format codes and rules are processed. It is good practice to increment sequence numbers by 10—to provide a range for the insertion of format codes and rules.

Rule Logic

The Rule Logic column is used to create multidimensional lookups and check rule expressions. Rule Logic columns are processed for reports only in #ModeRule or #ModeList mode. After a rule logic is processed for a rule in the check report, Oracle Hyperion Financial Data Quality Management, Enterprise Edition flags the rule as passing or failing.

Check Rule Condition Expressions

Check rule expressions are used primarily to validate target-system account balances when performing multidimensional lookups. The expressions return a True or False result.

For example, the following returns true (OK) if Cash (a target account) has a positive balance, and false (Error) when it does not:

```
|,,,YTD,<Entity Currency>,,Cash,[ICP None],[None],[None],[None],
[None],,,,,,,,,,,,,,>0
```

In this example, the check rule expression returns true (OK) when the value of Cash (a target account) plus \$1000 is greater or equals the value of AccruedTax (another target account), and false (Error) when it does not:

```
|,,,YTD,<Entity Currency>,,Cash,[ICP None],[None],[None],[None],
[None],,,,,,,,,,,,,,+1000=>|,,,YTD,<Entity Currency>,,AccruedTax,[ICP None],
[None],[None],[None],[None],,,,,,,,,,,,,|
```

Using the Rule Logic Editor to Create Check Rules

The Rule Logic Editor facilitates creation of check rules. It helps you develop rule logic and enables you to create rules from the Rule Logic Editor. You use the **Rule Logic Editor** to modify check rules.

To open the Rule Logic Editor:

1. On the **Setup** tab, under **Data Load Setup**, select **Check Rule Group**.
2. From **Check Rules**, in the **Check Rule Group** summary grid, select a check rule group.
3. From the **Rule Item Details** grid, click **Add**.
A row is added to the grid.
4. In each field, enter check rule information:
 - **Display Value**—See [Display Value](#).
 - **Description**—(optional) See [Description](#).
 - **Rule Name**—See [Rule Name](#).
 - **Rule Text**—See [Rule Text](#).
 - **Category**—See [Category](#).
 - **Sequence**—See [Sequence](#).
5. Click .

The **Rule Logic** screen includes three tabs:

- Rule Logic Add/Edit
- Rule Logic Add/Edit as Text
- Rule Logic Test Expression

Adding Rule Logic

Use the Rule Logic Add/Edit tab to add each row for the rule logic statement with a list of member values.

The Rule Logic Add/Edit tab consists of these elements:

- **Condition Summary**—Provides the logic for the check rule by enabling the specification of conditional expression that evaluates to "true" or "false."
If the Condition Summary does not contain a condition, it does not show **OK** or **Error**, but it does show an intersection that you can specify.
The Condition Summary and Display Summary options are optional, but if omitted, they display only 0.00.
- **Display Summary**—Enables the specification of a multiple-dimensional lookup to be used as the display value.
The display value is ignored and if the row has no rule logic at all and just a display value, the row is ignored and the report is terminated. The only value displayed in the Value column on the report is whatever expression is contained in the Display Summary. The display summary is optional.
- **Add**—Adds a row in the summary.
- **Delete**—Removes a row in the summary.

To add a rule logic statement:

1. On **Rule Logic Editor**, select the **Rule Logic Add/Edit** tab.
2. From the **Condition** or **Display** summary grid, click **Add**.
A blank line is displayed.
3. Enter the rule to test.

Note:

When using the equal sign for evaluating amounts, use double equal signs (==).

4. **Optional:** Click .
5. From **Rule Logic** in the **Intersection Type** field, select the intersection type for the multidimensional lookup.
Available intersection types:
 - Source intersection—Values are enclosed by the " ~" character.
 - Converted source intersection—Values are enclosed by the ' ' character.
 - Target intersection—Values are enclosed by the " |" character.
 See [Multidimensional Lookup](#).
6. From **Dimension**, select the dimension from which to retrieve values.
7. From **Member Value**, select a value from the dimension.

8. Click **Add to Intersection**.
The member value is added to the Display area.
9. Click **OK**.

Rule Logic Intersection

Use the **Rule Logic** Intersection screen, from which you can select dimensions directly from the target system, to ensure that required dimensions are entered and ordered correctly.

The screenshot shows the 'Rule Logic Intersection' dialog box. At the top, there is a title bar with a green checkmark icon and the text 'Rule Logic Intersection'. Below the title bar, there are two dropdown menus: 'Intersection Type' set to 'Target Intersection' and 'Dimension' set to 'Account'. Below these is a search bar with a 'Search' button. A list of member values is displayed, with 'A1_1000' selected. Below the list is an 'Add to Intersection' button. At the bottom, there is a large text area for the display, which currently shows a series of slashes '/////////////////'. At the bottom right, there are 'OK' and 'Cancel' buttons.

Member Value
A1_1000
A1_1100
A1_1110
A1_1120
A1_1130
A1_1200
A1_1211
A1_1212
A1_1213
A1_1220
A1_1222

Display the Rule Logic Intersection screen by clicking  from the Condition Summary or Display summary grid on the Rule Logic Add/Edit screen.

The Rule Logic Intersection screen enables you to select the type of retrieval format for the target dimensions.

Oracle Hyperion Financial Data Quality Management, Enterprise Edition uses the intersection type when multidimensional lookups are selected for a rules logic statement. The multidimensional lookup retrieves account values from the target system, FDMEE source data, target data or FDMEE source converted data. See [Multidimensional Lookup](#).

Multidimensional Lookup

The multidimensional lookup retrieves account values from the target system, Oracle Hyperion Financial Data Quality Management, Enterprise Edition source data, or FDMEE converted data. You can use multidimensional lookups in the rule condition and in the display of the rule logic.

Rule Data Sources

Oracle Hyperion Financial Data Quality Management, Enterprise Edition can retrieve data from three sources:

- Target-system data
- FDMEE source data
- FDMEE converted data

Target System Data

The following format, which begins and ends the rule with the pipe character (|), enables Oracle Hyperion Financial Data Quality Management, Enterprise Edition to retrieve target-system values for any dimension.

Unless otherwise specified, parameters are optional.

```
|Scenario, Period, Year, View, Value, Entity, Account (Required), ICP, Custom1,
Custom2,Custom3, Custom4, Custom5, Custom6, Custom7, Custom8, Custom9, Custom10,
Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18,
Custom19, Custom20|
```

The following examples illustrate ways that target-system values can be retrieved. In each example, Balance is a target account. For dimensions that are not referenced, you must use commas as placeholders.

Note the following:

- The Year dimension defaults to the year set in the POV.
- The Currency dimension defaults to 0.
- The View dimension defaults to YTD.
- The Value dimension defaults to <Entity Currency>.

Example 1

Look up the value of Balance for the target period and scenario (category) set in the POV and for each entity of the FDMEE check entity group that is assigned to the location. The example rule passes the check when the target account is less than \$10 and greater than -10.

```
|,,,,,Balance,,,,,,,,,,,,,| > -10.00 AND
|,,,,,Balance,,,,,,,,,,,,,| < 10.00
```


Example 2

Look up the value of Balance for the specified dimensions.

```
|Actual, March, 2002, YTD, Ohio, Balance, Michigan, Engines, Ford, Trucks,  
[None],,,,,,,,,,USD| > 0
```

Example 3

Look up the value of Balance for the specified dimensions and the previous period.

```
|Actual, -1, 2002, YTD, Ohio, Balance, Michigan, Engines, Ford, Trucks,  
[None],,,,,,,,,,USD| > 0
```

Example 4

Look up the value of Balance for the target scenario (category) set in the FDMEE POV, the previous target period, and each entity of the FDMEE check entity group that is assigned to the location.

Example 1

The following shows how to use +n and -n to specify a relative offset in the check rule when the current year dimension is "2015":

-1 result is 2015 - 1 = 2014 (Year - n)

+1 result is 2015 + 1 = 2016 (Year + n)

Example 2

The following shows how to use +n and -n to specify a relative offset in the check rule when the current period dimension is "January":

-1 result is January - 1 = January

+1 result is January + 1 = February

+12 result is January + 12 = December

FDMEE Source Data

The following format, which begins and ends the rule with the tilde character (~), retrieves values from data that was mapped to a target member, and then loaded into Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

Unless otherwise specified, parameters are optional. Parameters designated UD# are user-defined.

```
~FDMEE Category, FDMEE Period, Year (Field Not Applicable), FDMEE View, FDMEE  
Location, Source Entity(Required), Source Account(Required), Source ICP, Source  
UD1, Source UD2, Source UD3, Source UD4, Source UD5, Source UD6, Source UD7, Source  
UD8, Source UD9, Source UD10, Source UD11, Source UD12, Source UD13, Source UD14,  
Source UD15, Source UD16, Source UD17, Source UD18, Source UD19, Source UD20~
```

FDMEE Converted Data

The following format, which begins and ends the rule with the grave accent character (`), retrieves pull values from data that was loaded into Oracle Hyperion Financial Data Quality Management, Enterprise Edition. Unless otherwise specified, the parameters are optional.

```
`FDMEE Category, FDMEE Period, Year (Field Not Applicable), FDMEE View, FDMEE  
Location, Entity(Required), Account(Required), ICP, Custom1, Custom2, Custom3,
```

Custom4, Custom5, Custom6, Custom7, Custom8, Custom9, Custom10, Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18, Custom19, Custom20`

Math Operators

Math Operators (+, -, *, /)—If you select a math operator, then the check rule has an amount that equals the original amount calculated with the specified expression. For example, when you select the operator “*” and enter 2 in the rule field, then the new record is an amount two times the original amount. The math operators available in the expressions:

- + (addition)
- - (subtraction)
- * (multiplication)
- / (division)
- abs ()

If/Then/Else

Check rules accept If/Then/Else statements that enables you to create a more complex conditional tests on the Add/Edit as Text tab. This statement provides a primary path of execution when the if statement evaluates to "true," and a secondary path of execution when the if statement evaluates to "false."

Using the If/Then/Else statement, you can use custom-field values within reports as warning messages and flags.

In the following example, when the Result is between 100 to 1500, the Check Report with Warning prints "Amount between 100 and 1500." The example references three data accounts:

1. 24000050: 1000
2. 24000055: 500
3. 24000060: 10

This calculation for this example is $1000 + 500/10$, with the result of 1050.

The script is written using Jython code:

```
def runVal():

    dbVal=abs((|,,,,,BERLIN,24000050,[ICP None],[None],[None],[None],
[None],,,,,,,|)+(|,,,,,BERLIN,24000055,[ICP None],[None],[None],[None],
[None],,,,,,,|)/(|,,,,,BERLIN,24000060,[ICP None],[None],[None],[None],
[None],,,,,,,|))

    PstrCheckMessage1=''

msg2=''

msg3=''

    if(dbVal<100):

        RESULT=True
```



```

PstrCheckMessage1='Amount < 100.'

elif(dbVal>100 and dbVal<=1500):

    RESULT=True

    PstrCheckMessage1='Amount between 100 and 1500.'

elif(dbVal>1500 and dbVal<=9999):

    RESULT=True

    PstrCheckMessage1='Amount between 1501 and 9999.'

else:

    RESULT=False

    PstrCheckMessage1='Amount greater than 9999!'

return [RESULT,PstrCheckMessage1,msg2,msg3]

```

 Note:

You must include three message parameters with the return statement to write data to the status table. Regardless of whether you are only writing a single message, the other two message parameters are required.

The result of running this script is shown in the Check Report with Warnings:

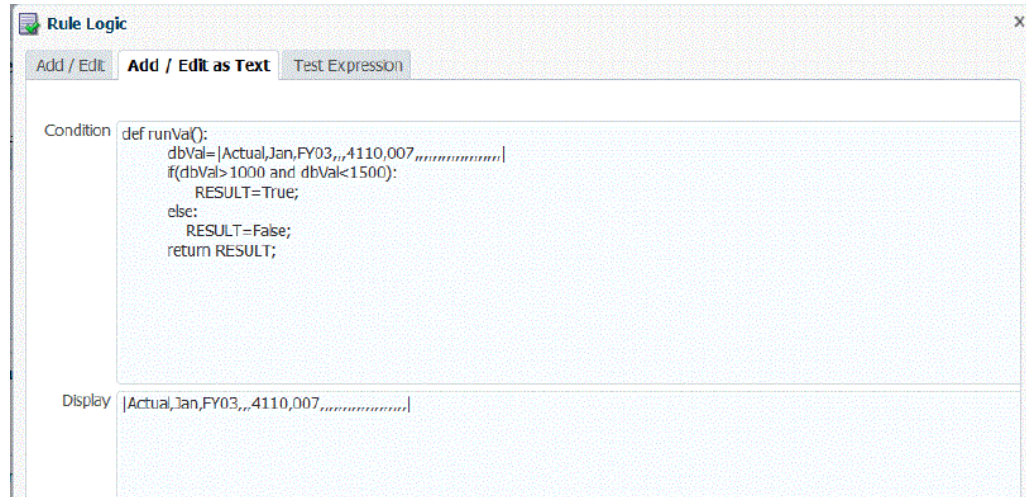
			Location: MultiPer_SSN
			Category: Actual
			Period: 2008-05-01
Validation Group: Check			
Pass			
BERLIN			
Account	Rule Definition		Value
OK	24000050	24000050	1,050.00
			Amount between 100 and 1500.

Adding a Rule Logic Statement as Free Form Text

Use the **Rule Logic Add/Edit as Text** tab to add the rule to test manually as free-form text. This feature enables you to instruct the system on how to display the actual value from the check rule logic. It always provides you with a way to cut and paste text rather than entering text line by line. If you do not specify a display summary, then the report shows a "0" value.

When the Display area is active, all interactions take place in the Display area. For example, pasting a tree copies all tree members to the Display area. When the Rule

area is active, all interactions take place in the Condition area. Changes made on the Lookup tab are reflected on the Rule tab in related editors.



To add a rule logic statement:

1. On the **Rule Logic Editor**, select the **Rule Logic Add/Edit as Text** tab.
2. In **Rule**, enter the rule.

Do not use a semicolon (;) in check rules. The semicolon is a reserved as the separator between the rule value and the display value.

When using the equal sign for evaluating amounts, use double equal signs (==) instead of the single equal sign (=). For example, use $a - b == 0$, and not $a - b = 0$.

3. Click **OK**.

Testing Check Rule Expressions

You can test a rule in the Test Expression tab of the Rule Logic tab and validate conditions for a combination of dimensions. When a test check rule is submitted, you can display errors that were found in the rule being tested.

The Text Expression screen contains the following elements:

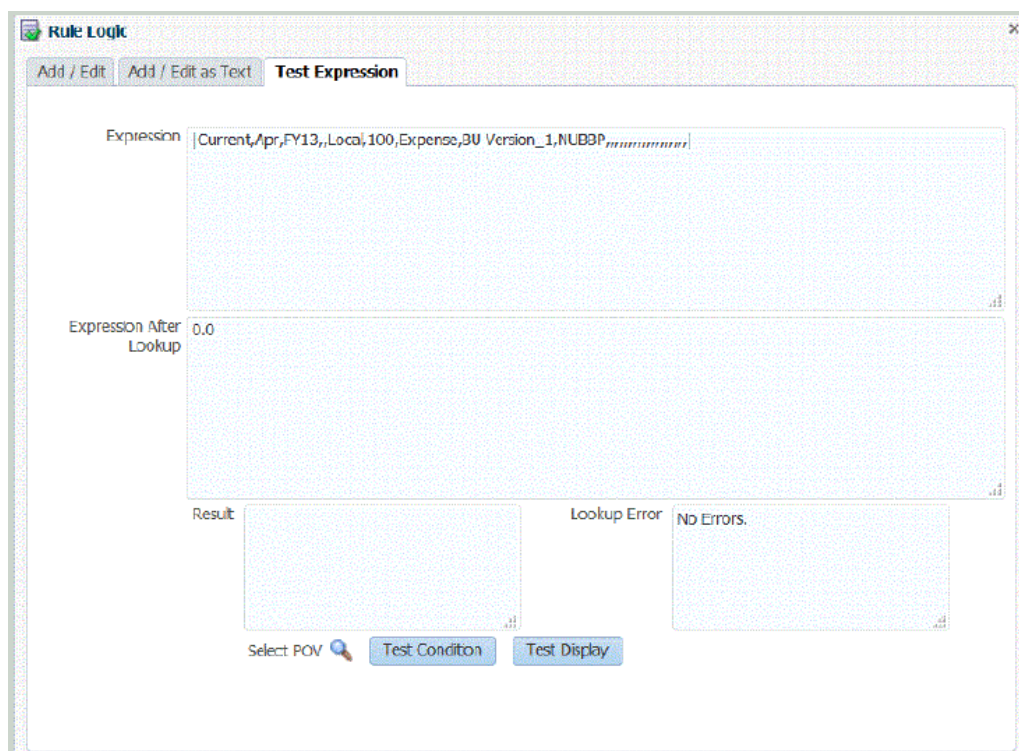
- **Expression**—Area that displays the rule being tested
- **Expression After Lookup (Scratch Pad)**—Area that displays the results of the expression being tested. (You can use the area for temporary storage of text and expressions by selecting text and right-clicking.)
Any expression in this field is removed when you click Test Condition or Test Display.
- **Result**—Area that displays the results of the test condition (True or False)
- **Lookup Errors**—Area that displays errors that were found in the expression being tested
- **Select POV**—Enables you to select the Check Rule POV.

On the Select Point of View screen, you can select another POV, or select a specific entity on the Member Selector screen.

 Note:

The POV can only be set after data was exported to the application for a specific POV. Then you can enter the POV and run the rule being tested. The POV entered remains set for the current session. You can navigate to the workbench and return without having to reset the POV.

- **Test Condition and Test Display**—Buttons that are used to run, respectively, the expression in the Condition or Display area on the Rule tab



To test a rule logic statement:

1. On the **Rule Logic Editor**, select the **Test Expression** tab.
2. Click **Select POV** to select the POV for the rule.
Optionally, you can click the **Entity** look up and select a member from the **Member Selector** screen.
3. Click **Test Condition** or **Test Display** to test the condition or display specified on the other tabs.
If you need to edit an expression, move to the other Rule tabs (Add Edit and Add / Edit as Text) and change the expression.
4. Click **OK**.

Running Check Reports for Profitability and Cost Management BSO Essbase Cube

Before running a Check Report for an Oracle Hyperion Profitability and Cost Management BSO Essbase cube, make sure that a fully qualified name is used to

differentiate any duplicate members used within dimensions or across dimensions. To construct a fully qualified name from a duplicate member name, add the fully qualified name as a rule logic statement on the Rule Logic Add/Edit as Text screen (see [Adding a Rule Logic Statement as Free Form Text](#)).

Note that a fully qualified member name comprises the duplicate member or alias name and all ancestors up to and including the dimension name. Each name must be enclosed in brackets ([]) and separated by a period (.). The syntax is as follows:

```
[DimensionMember].[Ancestors...].[DuplicateMember]
```

For example:

```
[Market].[East].[State].[New York]
```

```
[Market].[East].[City].[New York]
```

See *Oracle® Essbase Database Administrator's Guide*.

Creating Check Entity Groups

A check entity group consists of one or more target-system entities. When a check report is generated, the entities of the entity groups that are assigned to the report location are consolidated and displayed in the report. You activate check entity groups by assigning them to locations. You define the check entities of a check entity group by entering values in the fields of the check entities form of the **Check Entities** screen.

The Check Entity Group screen consists of three grids:

- Check Entity Group Summary—Summary area that lists the names of the check entity group, and enables you to create a new check entity group.
- Check Entity Group Details—Detail area where you can name and describe the check entity group.
- Entity Details—Detail area where you can add information about the entity.

To add a check entity group:

1. On the **Setup** tab, under **Data Load Setup**, select **Check Entity Group**.
2. On the **Check Entity Group** grid, click **Add**.
A blank row is added at the top of the grid.
3. From the Check Entity Group Detail area, enter the check entity group name in the **Name** field.
Optional: In the **Description** field, enter a description of the check entity group.
4. Click **Save**.

To add an entity detail:

1. In **Check Entity Group** summary grid, select a check entity group.
2. In **Entity** details grid, click **Add**.
Blank options lines are displayed.
3. Complete the following fields:
 - Parent

- Entity
- Consolidate
- On Report
- Sequence

4. Click **Save**.

Table 5-2 Entity Detail Options and Descriptions

Option	Description
Parent	Specify the organization in which the entity is consolidated. For other target systems, you select the parent of the entity. If the Consolidate option is not selected, the selection is irrelevant.
Script Name	For Planning and Essbase calculation scripts, specify the calculation script name to execute. This field is only available when the Check Entity Calculation method is set to "calculation script" in the Essbase or Planning application options.
Calc Script Parameters	Click  to browse and set the script for the calculation script on the Set Calculation Script Parameters screen. You can also add a calculation script on the Set Calculation Script Parameters screen. As part of the check rule process, Oracle Hyperion Financial Data Quality Management, Enterprise Edition references any calculation script parameters included in custom calculation scripts. A calculation script is a series of calculation commands, equations, and formulas that enable you to define calculations other than those defined by the database outline. Calculation scripts are defined in Essbase and Planning target application options, see Using Calculation Scripts. This field is only available when the Check Entity Calculation method is set to "calculation script" in the Essbase or Planning application options. If the Check Entity Calculation method is set to "dynamic", this field is unavailable.
Entity	Specify the target entity to consolidate and display in the check report. If the Consolidate option is selected, the entity is consolidated before it is displayed in the check report.

Table 5-2 (Cont.) Entity Detail Options and Descriptions

Option	Description
Consolidate	Select to consolidate an entity prior to displaying it in the check report. FDMEER also runs a consolidation after loading the target system (assuming a check entity group is assigned to the location). The consolidated entities are specified in the check entity group that is assigned to the active location. Oracle Hyperion Planning—Runs the default calculation or calculation script specified in the Calc Script Name depending on the "Check Entity Calculation Method" property of the target Application. Oracle Essbase—Runs the default calculation or calculation script specified in the Calc Script Name depending on the "Check Entity Calculation Method" property of the target Application. Oracle Hyperion Financial Management—The consolidation of data occurs in the Financial Management database.
On Report	The option selected in the On Report column determines whether an entity is displayed in the check report. If On Report is not selected and Consolidate is selected, the entity is consolidated but not displayed.
Sequence	Specify the order in which entities are consolidated and displayed in the check report. It is good practice to increment the sequence number by 10, to provide a range for the insertion of entities.

6

Batch Processing

Using the Oracle Hyperion Financial Data Quality Management, Enterprise Edition batch processing feature, you can:

- Combine one or more load rules in a batch and execute it at one time.
- Run jobs in a batch in serial or parallel mode.
- Define the parameters of the batch.
- Derive the period parameters based on POV settings.
- Create a "master" batch that includes multiple batches with different parameters.

For example, you can have one batch for metadata rules run in serial mode, and a second batch for the data rule run in parallel mode.

- Associate a batch with a batch group for ease of use and security.
- Instruct the batch to submit included jobs in parallel mode and return control.
- Instruct the batch to submit included jobs in parallel mode and return control only when *all* jobs are complete.

Batch processing options are available on the FDMEE task pane, or by executing batch scripts.

If you process batches from the FDMEE task pane, use the Batch Definition option to create a batch, and specify its parameters and tasks included in the batch. See [Working with Batch Definitions](#). Use the Batch Execution option to execute batches. See [Executing Batches](#).

Note:

FDMEE batch load features are unavailable to the Account Reconciliation Manager. For Accounts Reconciliation Manager load and scheduling features, see the *Oracle Hyperion Financial Close Management User's Guide*.

Working with Batch Definitions

A batch definition is used to define the batch jobs and parameters, and the type of rules included in the batch. A batch can contain one type of rule only. Valid types of rules are:

- metadata
- data
- batch
- open batch

**Note:**

Only an administrator can create batch definitions.

The Batch Definition features consist of three regions:

- **Batch Definition detail**—Enables you to add and delete a batch definition. If adding or modifying a definition, specify the definition name, target application, process method, return control method, and wait parameters.
- **Batch Definition parameters**—Enables you to derive period parameters based on the Import to Source, Export to Target, POV period, and to indicate data extract parameters. The parameter definition is unavailable for the batch types "batch" and "metadata (rules)."
- **Batch Definition jobs**—Enables you to add and delete jobs in a batch. Based on the type of batch, specific types of rules are allowed.

To add a batch definition:

1. On the **Setup** tab, under **Batch**, select **Batch Definition**.
2. In the **Batch Definition** summary section, click **Add**.
Use the blank Name and Target Application fields in the Batch Definition summary grid to enter a batch name or target application on which to search.
3. In **Batch Definition** detail section, select the **Definition** tab.
4. In **Name**, specify the name of the batch definition.
The name must contain only alpha, numeric or underscore characters. Do not enter spaces or any other character.
5. From **Target Application**, select the name of the target application.
6. From **Type**, select the type of rule for the definition.

Available types are:

- metadata
- data
- batch
- open batch—file based data sources
- open batch multi-period file-based data sources that include starting and ending periods.

The Open Batch type is used only for file-based data sources and does not contain any batch jobs. When you run this type of batch, the process reads the files automatically from the openbatch directory and imports them into the appropriate POV based on the file name. When the open batch is run, the master folder is emptied.

7. From **Execution Mode**, select the batch process method.
 - **Serial**—Processes files sequentially, requiring that one file complete its process before the next file starts its process.
 - **Parallel**—Processes files simultaneously.

 Note:

Files are not grouped by location in parallel mode.

8. For batch processing run in parallel mode, complete the following fields
 - **Wait for Completion**—Select **Wait** to return control only when the batch has finished processed.

Select *No Wait* to run the batch in the background. In this case, control is returned immediately.
 - **Timeout**—Specify the maximum time the job can run. Oracle Hyperion Financial Data Quality Management, Enterprise Edition waits for the job to complete before returning control.

The Timeout can be in seconds or minutes. Enter a number followed by a **S** for seconds or **M** for minutes.
9. In **Open Batch Directory** for an open batch type, specify the folder under Home \inbox\batches openbatch where the files to be imported are copied. If this field is blank or null, all files under Home\inbox\batches\openbatch are processed.
10. In **File Name Separator** for an open batch, select the character to use when separating the five segments of an open batch file name.
Options:
 - ~
 - @
 - ;
 - _
11. Select **Auto Create Data Rule** to create the data rule automatically for file-based data loads.

 Note:

The Auto Create Data Rule option is available when the rule type is “open batch.”

When FDMEE assigns the data rule name, it checks whether a data rule with the name "Location_Category" exists. If this name does not exist, FDMEE creates the data rule using the following file naming conventions:

- Rule Name—Location_Category
 - Description—“Auto created data rule”
 - Category—Category
 - File Name—Null
 - Mode—Replace
12. **Optional:** In the **Description** field, enter a description of the batch definition.
 13. Click **Save**.

- 14. Optional:** In **Batch Group**, select the batch group to associate with the batch.

For more information, see [Adding a Batch Group](#).

- 15. Optional:** In **Number of Parallel Jobs**, specify the maximum number of parallel processes submitted by a batch at any time.

This option is used in conjunction with the **Wait for Completion** and **Timeout** fields.

If Wait for Completion and a time-out period options are set, but the number of parallel jobs is not, then FDMEE waits for all batch jobs to complete and then returns control.

When the number of parallel jobs count is set and Wait for Completion/time-out modes are enabled, the system submits the specified number of jobs for processing at one time. If the wait time is reached before all the jobs are complete, the system exits the batch processing procedure.

The Wait for Completion setting is not for each subset, but for the whole batch. For example, let's suppose you have 20 jobs where the number of parallel jobs is set to 4 and the time-out period is 10 M. If only 15 jobs are completed in 10 M, the system still exits.

If No Wait is specified, the system submits all jobs and returns control submitted immediately without waiting for any running processes to finish.

- 16. Optional:** In **Before Batch Script**, enter the name of the custom script to run before the batch.

You can browse and select, or upload a custom script on the Select screen by clicking .

 Note:

If the custom script is run in a "Before Batch Script" attached to a batch definition, store parameters in a custom table or file so that the parameters can be read.

The custom script can be written in Jython or Visual Basic script.

- 17. Optional:** In **After Batch Script**, enter the name of the custom script to run after the batch.

You can browse and select, or upload a custom script on the Select screen by clicking .

The custom script can be written in Jython or Visual Basic script.

- 18. Click **Save**.**

To add batch definition parameters:

1. On the **Setup** tab, under **Batch**, select **Batch Definition**.
2. In **Batch Definition**, under **Batch Definition Detail**, select the **Parameter** tab.
3. In **Parameters**, select **Import From Source** to import the data from the source system, perform the necessary transformations, and export the data to the FDMEE staging table.

4. Select **Export To Target** to export the data to the target application.
5. Select **POV Period** to derive the period parameters based on POV settings.

If the POV period is selected, the period key start and end period fields are disabled.

This field is only available for a data load batch.

When setting up a batch, you can choose the POV to drive the period, or enter the periods explicitly. If you select the POV, it is pulled from the Default POV Period setting in System/Application, or User settings.

6. Specify dates in the **Start Period** and **End Period** to derive period parameters through which the data is processed.

Use the date format based on the locale settings for your locale. For example, in the United States, enter the date using the format MM/DD/YY format.

If the Start Period and End Period fields are selected, the POV Period field is disabled.

This field is only available for a data load batch.

7. In the **Import Mode** drop-down, select the mode to extract data all at once for an entire period or incrementally during the period.

Option are:

- **Snapshot**—Extracts everything for the selected source set for an entire period.

Note the following behavior of Snapshot mode:

- When the source data for the selected period *has never been run*, data is extracted from the source.
- When the source data for the selected period *has been run*, data is extracted from the FDMEE staging table, and not from the source. This means that if a user has two locations that extract from the same Enterprise Resource Planning (ERP) source, data is extracted from the source only once (and the first time only).

For example, if you load data to Oracle Hyperion Financial Management from the Oracle E-Business Suite for a selected period, but you run the integration to ARM for the same source and period, use what is in the interface tables since data is only extracted the first time from the source. This result in a significant performance gain for any subsequent data loads. The first extraction take the longest, but any other subsequent extractions are fast.

- **Incremental**—Extracts those records that were added after the prior data extract

 Note:

When using incremental mode to pull data from Peoplesoft, note that the system determines the records to pull based on the most recent PROCESS_INSTANCE entry on the Peoplesoft General Ledger tables. For example, if a job is scheduled at 8:00 am to run at noon, and a user executes a job at 10 am, then only the records posted at 10 am are pulled for an incremental run. If records are not pulled over using incremental mode, then the user should run a full refresh to get all records.

- **Full Refresh**—Performs a clean extraction from the source system, thereby clearing any existing data rows in the appropriate FDMEE staging tables for a given source Ledger (or Business Unit) and source period.

When defining the file name for an open batch that uses multi-periods, specify the starting and ending periods in the file name, for example,
10~Filerule~Jan03~Mar03~FR.txt.

When defining the file for an open batch that uses a single period, specify the period in the file name, for example, 10~0BFilerule~Jan03~FR.txt.

 Note:

The import mode options (Snapshot, Incremental and Full Refresh) are only applicable to Data Rules in a Location using a Standard Import Format. Data Rules in a Location with a Source Adapter Import format always perform a full data extraction (similar to Full Refresh) directly into the TDATESEG_T table.

- **Append**—Existing rows for the POV remain the same, but new rows are appended to the POV. For example, a first time load has 100 rows and second load has 50 rows. In this case, FDMEE appends the 50 rows. After this load, the row total for the POV is 150.
- **Replace**—Replaces the rows in the POV with the rows in the load file (that is, replaces the rows in TDATESEG). For example, a first time load has 100 rows, and a second load has 70 rows. In this case, FDMEE first removes the 100 rows, and loads the 70 rows to TDATESEG. After this load, the row total for the POV is 70.

 Note:

If you run data load in Full Refresh mode in the Account Reconciliation Manager (ARM), select all locations that have data. Otherwise, FDMEE contains data for locations not selected in ARM (but ARM does not). This results in a discrepancy between what is in FDMEE and what is in ARM.



Note:

E-Business Suite and FUSION source imports require a full refresh of data load rules before export after upgrading from an 11.1.2.2 release.

This field is only available for a data load batch.

8. Select **Extract Exchange Rate** to extract the exchange rate.

See [How Currencies Are Processed](#).

(This option is not applicable for file-based source systems).

9. For Oracle Essbase, Oracle Hyperion Planning, and Options for the Financial Management applications, in the **Export Mode** drop-down, select the mode of exporting data.

Options for Essbase or Planning applications:

- STORE_DATA
- REPLACE_DATA
- ADD_DATA
- SUBTRACT_DATA
- OVERRIDE_ALL_DATA

This field is only available for a data load batch.

Options for the Financial Management applications:

- Accumulate
- Replace
- Merge
- Replace by Security

10. For Essbase or Planning, from the **Plan Type** drop-down, select the plan type of the application.

11. Click **Save**.

To add a batch job:

1. On the **Setup** tab, under **Batch**, select **Batch Definition**.
2. In **Batch Definition**, under **Batch Jobs**, click **Add**.

In **Rule Name**, specify the rule name associated with the batch job.



You can also select the  to navigate to and select the rule name.

3. In **Job Sequence**, specify the order in which to sequence the batch.
4. Click **Save**.

Adding a Batch Group

Batch groups enable you to determine security eligibility when executing batches.

See [Setting Security Options](#).

To add a batch group:

1. On the **Setup** tab, under **Batch**, select **Batch Definition**.
2. Select the **Batch Group** tab.
3. Click **Add**.
4. In **Name**, specify the batch group name.
5. Click **Save**.
6. **Optional:** Associate a batch with a batch group:
 - a. Select the **Batch** tab.
 - b. From the **Batch** summary section, select the batch to associate with a batch group.
 - c. From the **Batch** detail section, select the **Definition** tab.
 - d. In **Batch Group**, select the batch group to associate with the batch.
 - e. Click **Save**.

Executing Batches

Use the Batch Execution feature to show all batches that you have accessed based on the batch group assigned. You can also use the Batch Execution feature to select a batch and execute a rule after parameters passed with the rule have been validated.

Batch Execution shows all batches to which you have access based on the batch group assigned.



Note:

The Batch Execution option is only accessible to a user with a Run Integration role.

To execute a rule:

1. On the **Workflow** tab, under **Other**, select **Batch Execution**.
2. In the **Batch Execution** summary area, select a batch name, and then click **Execute**.
3. **Optional:** You can also schedule a job by clicking **Schedule** (see [Scheduling Jobs](#)). You can check the status of the batch by clicking **Check Status** (see [Viewing Process Details](#)).

Using Open Batches

The open batch functionality is used to read file-based data sources and import them into the appropriate POV based on the file name. It is a batch definition type that enables you to automate and schedule the workflow process (such as four fish steps in the Data Load Workbench: import, validate, export, and check), particularly when

you have a need to process the loading large number of external files. Open batches cannot contain jobs. Additionally, open batches can be scheduled to run periodically.

The high-level process overview of the Open Batches function consists of:

1. In **Batch Definition**, add a new batch definition with the type of **Open Batch**.
2. Create an openbatch folder in the application inbox\batches subdirectory where the files to be imported are copied.

After a batch is processed, a directory is created and all files within the OpenBatch directory are moved into it. The new directory is assigned a unique batch ID.
3. Select the **File Name Separator** character.

This character is used to separate the five segments of an open batch file name.
4. Select the **Auto Create Data Rule** option.
5. Stage the open batch files by copying files to the inbox\batches\openbatch folder using the name format for batch files.
6. In Batch Execution, process the batch.

Name Format for Open Batch Files

The names of open batch files consist of the following segments:

- POV—Point-of-View where Oracle Hyperion Financial Data Quality Management, Enterprise Edition processes the data, such as location, period, and category
- Import load method—how FDMEE imports data to the POV
- Export load method—how FDMEE exports data to the target application

Open batches are located in the inbox\batches\openbatch, and under a sub-folder if one has been created for them.

Naming Conventions

Oracle Hyperion Financial Data Quality Management, Enterprise Edition uses the following methods to name open batch files:

1. Auto Create Data Load Rule—used when the data load rule is created the first time.
2. Existing Data Load Rule—used when the data load rule has been created in FDMEE.

When FDMEE creates the data rule name automatically, it checks whether a data rule with the " Location_Category " name exists. If the name does not exist, FDMEE creates the data rule using the file naming convention:FileID~Location~Category~Period~RR.ext (this example uses a ~ (tilde) file name separator).

- File ID—a free-form value used to sort the file for a given Location. Files are sorted by Location and within a location, the File ID is used to sort the files.
- Location—POV location
- Category—POV Category
- Period—POV Period

- load method—a two character value. The first character indicates the import Format, and the second character indicates the export format.

Valid values for import mode are **A** and **R**.

Valid values for export mode are **A**, **R**, **M**, and **S**.

For more information, see [Open Batch Import and Export Load Methods](#).

- ext—file extension (txt or csv)

When FDMEE finds that a data rule already exists for an open batch, it uses the file naming convention: FileID~Period~RR.ext (this example uses a ~ (tilde) file name separator)

- File ID—a free-form value used to define the load order of batch execution. Batch files load in alphabetic order by file name.
- Data Load Rule—Data Load Rule name
- Period—POV Period
- load method—a two character value. The first character indicates the import Format, and the second character indicates the export format.

Valid values for import mode are **A** and **R**.

Valid values for export mode are **A**, **R**, **M**, and **S**.

For more information, see [Open Batch Import and Export Load Methods](#).

- ext—file extension (txt or csv)

Open Batch Import and Export Load Methods

Import load methods include:

Import Load Method	File-Based	Fusion/E-Business Suite/ PeopleSoft
A	Append	not applicable
R	Replace	not applicable
F	not applicable	Full Refresh
I	not applicable	Incremental
S	not applicable	Snapshot

Export load methods include:

Export Load Method	Hyperion Financial Management	Planning/Essbase/Hyperion Profitability
A (ADD_DATA)	Accumulate	not applicable
R	Replace	Applicable for Planning only
M (STORE_DATA)	Merge	Merge for Essbase, Full Refresh for Hyperion Profitability. (The merge export load method is not available for Planning).
S	Replace by Security	Subtract for Planning, Replace for Essbase, Incremental for Hyperion Profitability

Export load methods include:

Creating Open Batches

To create and process open batch:

1. On the **Setup** tab, under **Batch**, select **Batch Definition**.
2. In the **Batch Definition** summary section, click **Add**.
Use the blank Name and Target Application fields in the Batch Definition summary grid to enter a batch name or target application on which to search.
3. In **Batch Definition Detail** section, select the **Definition** tab.
4. In **Name**, specify the name of the batch definition.
5. From the **Target Application** drop-down, select the name of the target application.
6. In the **Type** drop-down, select **Open Batch**.
7. In the **Execution Mode** drop-down, select **Serial**.
The serial execution mode processes files sequentially, requiring that one file complete its process before the next file starts its process.
8. In **Open Batch Directory**, specify the folder under `inbox\batches\openbatch` subdirectory where the files to be imported are copied. If this field is blank or null, all files under `inbox\batches\openbatch` are processed.
9. In **File Name Separator**, for an open batch type, select the character to use when separating the five segments of a batch file name.
Options:
 - ~
 - @
 - _
 - ;
10. Select **Auto Create Data Rule** to create the data rule automatically for file-based data loads.
When Oracle Hyperion Financial Data Quality Management, Enterprise Edition assigns the data rule name, it checks whether a data rule with the "Location_Category" name exists. If the name does not exist, FDMEE creates the data rule.
To use predefined data rules that load data based on specific categories, leave this field blank.
11. **Optional:** In the **Description** field, enter a description of the batch definition.
12. Click **Save**.
13. Stage the file-based data source files by copying them to `inbox\batches\openbatch` using one of the following methods:
 - **Predefined Data Load Rule**—To use a predefined data rule that loads data based on specific categories, leave the Auto Create Data Rule field blank on the Batch Definition screen and create the data load rule (see [Defining Data Load Rules to Extract Data](#)).

Next, create the open batch file name using the following format:

FileID_RuleName_Period_LoadMethod. The file id is a free-form field that you can use to control the load order. Batch files load in alphabetic order by file name.

The load method is defined using two-character code identifying the load method where the first code represents the append or replace method for the source load, and second character represents the accumulate or replace method for the target load.

For the import load method, depending on the source system, available values are:

- A—Append
- R—Replace Data
- F—Full Refresh
- I—Incremental
- S—Replace by Security

For Oracle Hyperion Financial Management the export load method are:

- A—Accumulate
- R—Replace
- M—Merge
- S—Replace by Security

For Oracle Hyperion Planning, the export modes are:

- A—ADD_DATA
- R—REPLACE_DATA
- M—STORE_DATA
- S—SUBTRACT_DATA

ExaOracle Tax Reporting Cloudmplies of an open batch file name are:

a_Texas_Actual04_Jan-2004_RR.txt and b_Texas_Actual04_Jan-2004_RR.txt

- Auto-Created Data Load Rule—To load data to any location category and have FDMEE create the data load rule automatically, create the open batch file name using the following format:
“FileID_Location_Category_Period_LoadMethod”.

In this case, FDMEE looks for the data rule with the name “Location_Category.” If it does not exist, FDMEE creates the data rule automatically with the name “Location_Category.”

14. **Optional:** Apply any scheduling conditions to the open batch file.

See [Scheduling Jobs](#).

15. On the **Workflow** tab, under **Other**, select **Batch Execution**.

16. In the **Batch Execution** summary area, select the open batch file, and then click **Execute**.

After an open batch is processed, a directory is created, all files within the openbatch directory are moved into the new directory, and the new directory is assigned a unique batch ID.

**Note:**

The Open batch feature is unavailable for the Account Reconciliation Manager.

Creating an Open Batch to Run an Integration with E-Business Suite

You can use Open Batch functionality to run an integration with Oracle E-Business Suite. To do this, you create an empty file with the POV and the data load rule in the file name, and then save it to the open batch folder on the server. When you run the open batch process, Oracle Hyperion Financial Data Quality Management, Enterprise Edition runs the E-Business Suite integration for the specified rule and POV.

Creating Open Batches for Multiple Periods

You can use the open batch functionality to read file-based data sources with multiple periods and import them into the appropriate POV based on the file name. This feature enables you to automate the process of loading large number of files. Open batches for multiple periods cannot contain jobs. Additionally, open batches for multiple periods can be scheduled to run periodically.

Files for an open batch multiple period load are stored in the `inbox\batches\openbatchm1` directory.

The names of multiple period batch files consist of the following segments in the following order:

- File ID—A free-form field used to control load order. Batch files load in alphabetic order by file name.
- Location
- Category
- Start Period
- End Period
- Load Method—A two-character item (Character 1 = append or replace, and Character 2 = target append or replace). Valid values are **A** and **R**.

Examples of open batch for a multiple period file name:

`a_Texas_Actual_ Jan-2004_ Jun-2004_RR.txt` (Loc, Cat, Start Period, End Period)

and

`b_TexasDR1_ Jan-2004_ Jun-2004_RR.txt` (Data Rule, Start Period, End Period)

To create and process an open batch:

1. On the **Setup** tab, under **Batch**, select **Batch Definition**.
2. In the **Batch Definition** summary area, click **Add**.

Use the blank Name and Target Application fields in the Batch Definition summary grid to enter a batch name or target application on which to search.

3. In **Batch Definition Detail** section, select the **Definition** tab.

4. In **Name**, specify the name of the batch definition.
5. From **Target Application**, select the name of the target application.
6. From **Type**, select **Open Batch Multi-Period**.
7. In the **Execution Mode** drop-down, select **Serial**.

The serial execution mode processes files sequentially, requiring that one file complete its process before the next file starts to process.
8. In **Open Batch Directory**, specify the folder under: `inbox\batches\openbatchml` subdirectory where the files to be imported are copied. If this field is blank or null, all files under: `inbox\batches\openbatchml` are processed.
9. In **File Name Separator**, for an open batch type, select the character to use when separating the five segments of a batch file name.

Options:

- ~
 - @
 - ;
 - _
10. Select **Auto Create Data Rule** to create the data rule automatically for file-based data loads.

When Oracle Hyperion Financial Data Quality Management, Enterprise Edition assigns the data rule name, it checks whether a data rule with the "Location_Category" name exists. If the name does not exist, FDMEE creates the data rule using the following file name convention:

- Rule Name–Location_Category
- Description—"Auto-created data rule"
- Category–Category
- File Name–Null
- Mode–Replace

To use predefined data rules that load data based on specific categories, leave this field blank

11. **Optional:** In the **Description** field, enter a description of the batch definition.
12. Click **Save**.
13. Stage the file-based data source files by copying them to `inbox\batches\openbatch` using one of the following methods:
 - **Predefined Data Load Rule**—To use a predefined data rule that loads data based on specific categories, leave the Auto Create Data Rule field blank on the Batch Definition screen and create the data load rule (see [Defining Data Load Rules to Extract Data](#)).

If you have to load to noncontiguous periods in the open batch, create the data rule in which the source period mapping are defined, and use this option.

Next, create the open batch file name using the following format:

FileID_RuleName_Period_LoadMethod. The file id is a free-form field that you can use to control the load order. Batch files load in alphabetic order by file name.

The load method is defined using two-character code identifying the load method where the first code represents the append or replace method for the source load, and second character represents the accumulate or replace method for the target load.

For the source load method, available values are:

- A—Append
- R—Replace

For the target load method, available values are:

- A—Accumulate
- R—Replace

Examples of an open batch file name are: a_Texas_Actual04_Jan-2004_RR.txt and b_Texas_Actual04_Jan-2004_RR.txt

- Auto-Created Data Load Rule—To load data to any location category and have FDMEE create the data load rule automatically, create the open batch file name using the format: “FileID_Location_Category_Period_LoadMethod.”

In this case, FDMEE looks for the data rule with the name “Location_Category.” If it does not exist, FDMEE creates the data rule automatically with the name “Location_Category.”

An auto-create data rule is only applicable for contiguous period loads. To load to noncontiguous periods, create the data rule in which the source period mappings are defined.

14. **Optional:** Apply any scheduling conditions to the open batch file.

See [Scheduling Jobs](#).

15. On the **Workflow** tab, under **Other**, select **Batch Execution**.

16. In the **Batch Execution** summary area, select an open batch file, and then click **Execute**.

After an open batch is processed, a directory is created and all files within the openbatch directory are moved to it. The new directory is assigned a unique batch ID.

**Note:**

The Open batch feature is unavailable for the Account Reconciliation Manager.

Scheduling Jobs

The scheduling jobs feature provides a method to orchestrate the execution times of metadata load rules and data load rules.

**Note:**

When you cancel a job from the Oracle Hyperion Financial Data Quality Management, Enterprise Edition user interface, all instances of a schedule for the object selected are cancelled. To cancel a specific instance of a schedule, cancel the job from the ODI studio or ODI console.

To schedule a job:

1. From the **Batch Execution** screen, **Metadata** screen, or **Data Load Rule** screen, select the batch name (from the Batch Execution screen) or rule (from the Metadata screen or Data Load Rule screens) to schedule and click **Schedule**.
2. In **Schedule**, select any rule feature specific options.
For example, if you select the **Schedule** option from the **Data Load Rule** screen, specify the Import from Source, Recalculate, Export to Target, and so on options.
3. Specify the type of scheduling and select the associated date and time parameters.
See [Table 1](#).
4. Click **OK**.

Table 6-1 Schedule Types and Parameters

Schedule Type	Data and Time Parameters
Simple	Submits the job for execution at a specific day and time, but does not repeat: Available options: • Timezone • Date • Hour(s) • Minute(s) • Second(s) • Select (AM/PM)

Table 6-1 (Cont.) Schedule Types and Parameters

Schedule Type	Data and Time Parameters
Hourly	Executes at the specified minutes and seconds after the top of the hour every hour until cancelled. Available options: • Timezone • Minute(s) • Second(s)
Daily	Executes at the same time each day. Available options: • Timezone • Hour(s) • Minute(s) • Second(s) • Select (AM/PM)
Weekly	Executes at the specific time for each day selected. Available options: • Timezone • Monday-Sunday • Hour(s) • Minute(s) • Second(s) • Select (AM/PM)
Monthly (day of month)	Execute at the specific time for the day of the month selected. Also enables you to select the "Last Day of Month" or "Day Before End of Month." Available options: • Timezone • Monthly Date • Hour(s) • Minute(s) • Second(s) • Select (AM/PM)

Table 6-1 (Cont.) Schedule Types and Parameters

Schedule Type	Data and Time Parameters
Monthly (week day)	You can select the first, second, third, fourth, fifth, last, and then the specific day or the week on which to execute the job. Available options: • Day of Month • Day • Hour(s) • Minute(s) • Second(s) • Select (AM/PM)

 Note:

The Timezone option is unavailable for the Monthly (week day) schedule type.

Canceling a Scheduled Job

When you cancel a scheduled job from Oracle Hyperion Financial Data Quality Management, Enterprise Edition, all instances of the scheduled job for the object selected are cancelled. To cancel a specific instance of a schedule, use the ODI Studio or ODI Console.

To can a scheduled job:

1. On the **Batch Execution** screen, select the batch.
2. Click **Cancel Schedule**.

Working with Batch Scripts

Oracle Hyperion Financial Data Quality Management, Enterprise Edition provides a set of Windows batch/UNIX shell scripts that enable users to execute data load rules from a command line or schedule loads from any scheduler without writing Java code. Batch scripts can be invoked from the command line. In turn, scripts call the data load and metadata load API in the FDMEE application server that execute the rules using the normal process used in data rule and workbench. Batch scripts are located under a <EPM_ORACLE_INSTANCE>/FinancialDataQuality directory (<EPM_ORACLE_INSTANCE> is typically is located at: C:\oracle\Middleware\user_projects\epmsystem1).

Using a batch script to run data load rules includes:

- Executing the batch script for data load rules. See [Executing the Batch Script for Data Load Rules](#).

- Executing the batch script for metadata rules.

 Note:

Period names cannot include spaces if used in a batch script.

Using Password Encryption

Oracle Hyperion Financial Data Quality Management, Enterprise Edition provides a Win/UNIX script to encrypt a password and store in a file. Script `encryptpassword.bat/sh` is located in `EPM ORACLE INSTANCE/FinancialDataQuality` directory.

To encrypt a password:

1. Navigate to the directory that has the batch files.
Typically, the batch file directory is `EPM ORACLE INSTANCE/FinancialDataQuality` directory
2. From the `EPM ORACLE INSTANCE/FinancialDataQuality` directory, at a command prompt, run the script **`encryptpassword.bat <passwordfile>`**.
3. When prompted, enter the password, and then click **Enter**.

Note the following:

- The password is masked when you type it.
- When running the batch script, you can provide the password by file name as a parameter in the following format: `-f:passwordfile`. The file name used as a parameter is placed in the location defined in “Encrypted Password Folder” option of System Settings.
- Do not include a disk or directory specification as part of the file name, just enter a file name with or without an extension.
- Replace the [file] with the actual file name, and do not include the brackets.
- The script encrypts the password and writes it to the file provided as a parameter.
- For information on running a data load rule batch script with a password from a file, see [Executing the Batch Script for Data Load Rules](#).
- For information on running a metadata load rule batch script with a password from a file, see [Executing the Batch Script for Metadata Rules](#).

Executing the Batch Script for Data Load Rules

To run the data load rule batch script with a plain text password:

1. Display a Windows command window or UNIX shell.
2. Paste and run the following command: `loaddata USER PASSWORD RULE_NAME IMPORT_FROM_SOURCE EXPORT_TO_TARGET EXPORT_MODE IMPORT_MODE LOAD_FX_RATE START_PERIOD_NAME END_PERIOD_NAME SYNC_MODE`

To run the data load rule batch script with a password from a file:

1. Display a Windows command window or UNIX shell.
2. Paste and run the following command: `loaddata USER -f:PASSWORD_FILE RULE_NAME
IMPORT_FROM_SOURCE EXPORT_TO_TARGET EXPORT_MORE IMPORT_MODE LOAD_FX_RATE
START_PERIOD_NAME END_PERIOD_NAME SYNC_MODE`

Setting the Parameters for Data Load Rules

The parameters used to execute a batch script for data load rules are:

Table 6-2 Parameters for Data Load Rules

Parameter	Value
User	Username
Password	Password or -f:Password file name
IMPORT_FROM_SOURCE	Y or N
EXPORT_TO_TARGET	Y or N
EXPORT_MODE	Oracle Essbase and Oracle Hyperion Planning applications export modes: • STORE_DATA • ADD_DATA • SUBTRACT_DATA • REPLACE_DATA • OVERRIDE_ALL_DATA The Oracle Hyperion Financial Management application export mode: • Accumulate • Replace • Merge • Replace_By_Security
IMPORT_MODE	• Snapshot • Incremental • Full Refresh The file-based source system export modes: • Append • Replace
LOAD_FX_RATE	Load exchange rate flag. Y or N

 Note:

Oracle E-Business Suite and FUSION source imports require a full refresh of data load rules before export after upgrading from an 11.1.2.2 release.

Table 6-2 (Cont.) Parameters for Data Load Rules

Parameter	Value
START_PERIOD_NAME	Period Name or POV if the POV specified period value is retrieved from the period profile.
END_PERIOD_NAME	Period Name or POV if the POV specified period value is retrieved from the period profile.
SYNC_MODE	SYNC/ASYNCC • SYNC—Process runs immediately and control returns when process completes. • ASYNC—When the ODI job is submitted, control returns. The load process continues to execute in ODI.

Executing the Batch Script for Metadata Rules

To run the metadata load rule batch script with a plain text password:

1. Display a Windows command window or UNIX shell.
2. At a Windows command window or UNIX shell, paste and run the following command: `loadmetadata USER PASSWORD LOCATION_NAME SYNC_MODE`

To run the metadata load rule batch script with a password from a file:

1. Display a Windows command window or UNIX shell.
2. At a Windows command window or UNIX shell, paste and run the following command: `loadmetadata USER -f:PASSWORD_FILE LOCATION_NAME SYNC_MODE`

Setting the Parameters for Metadata Rules

The parameters used to execute a batch script for metadata rules are:

Table 6-3 Parameters for Metadata Rules

Parameter	Value
User	Username
Password	Password or -f:Password file name
Location	Location Name
SYNC_MODE	SYNC/ASYNCC • SYNC—Process run immediately and control returns when the process completes. • ASYNC—When ODI job is submitted, control returns. The load process continues to execute executed in ODI.

Executing the Batch Script for HR Data Rules

To run the HR data rule batch script with a plain text password:

1. Display a Windows command window or UNIX shell.
2. Paste and run the following command: `loadhrdata USER PASSWORD LOCATION_NAME SYNC_MODE`

To run the HR data d rule batch script with a password from a file:

1. Display a Windows command window or UNIX shell.
2. Paste and run the following command: `loadhrdata USER -f:PASSWORD_FILE LOCATION_NAME SYNC_MODE.`

Setting the Parameters for HR Data Rules

The parameters used to execute a batch script for HR data rules are:

Table 6-4 Parameters for Human Resources Rules

Parameter	Value
User	Username
Password	Password or –f:Password file name
Rule Name	HR data rule name
IMPORT_FROM_SOURCE	Y or N
EXPORT_TO_TARGET	Y or N
As of Date	Date used to determine applicable effective date. Date must be in the YYYY-MM-DD format.
Load Type	Specify the load type: • Data—Loads only the data • Metadata—Loads only the Public Sector Planning and Budgeting metadata • Both—Loads the data and metadata.

Executing the Batch Script to Import Mapping Rules

You can use the `importmapping` batch script to import mappings rules from a command line.

To run the import mapping rule batch script with a password from a file:

1. Display a Windows command window or UNIX shell.
2. Paste and run the following command: `importmapping USER_NAME ENCY_PASS LOCATION_NAME DIMENSION_NAME FILE_PATH REPLACE VALIDATE SYNC_MODE`

Setting the Parameters to Import Mappings Rules

The parameters used to execute a batch script for importing mapping rules:

Table 6-5 Parameters for Data Load Rules

Parameter	Value
USER_NAME	Username
ENCY_PASS	Password or –f:Password file name

Table 6-5 (Cont.) Parameters for Data Load Rules

Parameter	Value
LOCATION_NAME	Location Name
DIMENSION_NAME	Name of the dimension
File_PATH	Directory from which to import source files.
Replace	Import mode load method: • A—Append • R—Replace
Validate	Y or N
SYNC_MODE	SYNC/ASYNCC • SYNC—Process runs immediately and control returns when process completes. • ASYNC—When the ODI job is submitted, control returns. The load process continues to execute in ODI.

 Note:

When using the importmapping.sh utility on Linux to import Oracle Hyperion Financial Data Quality Management, Enterprise Edition mappings, and you receive the following error: "String index out of range: -1", reference the directory like this: `./importmapping.sh admin welcome1 EBS_HFM_LOC account \app\EPM\import\FDMEE\FDMEE_Mapping_Account-PROJMGN.txt REPLACE N SYNC` . Then make sure that the file is in Unix format and not in windows format. Run `dos2unix` to convert the file and then reload.

Executing the Batch Script for Data Load Rules to Write Back

Use the "Loaddata" script to extract data from source EPM applications to target Enterprise Resource Planning (ERP) systems. See [Executing the Batch Script for Data Load Rules](#).

Running a Batch

To run the batch with a plain text password:

1. Display a Windows command window or UNIX shell.
2. Paste and run the following command: `runbatch USER PASSWORD BATCH_NAME SYNC_MODE`.

To run the batch with a password from a file:

1. Display a Windows command window or UNIX shell.
2. Paste and run the following command: `runbatch USER -f:PASSWORD_FILE BATCH_NAME SYNC_MODE`.

Setting the Parameters for Running the Batch

The parameters used to run the batch are:

Table 6-6 Parameters for running a batch.

Parameter	Value
User	Username
Password	Password or -f:Password file name
Rule Name	Batch Name

7

Creating and Using Scripts

Related Topics

- [Overview](#)
- [Key Jython Concepts](#)
- [Using the Script Editor](#)
- [Using Import Scripts](#)
- [Using Mapping Scripts](#)
- [Using Event Scripts](#)
- [Using Custom Scripts](#)
- [Using the JAVA API](#)
- [Using Visual Basic API and Samples](#)

Overview

Oracle Hyperion Financial Data Quality Management, Enterprise Edition offers a powerful extensibility framework by providing Jython and Visual Basic based scripting. Using the rich set of the API library, users can extend and customize the product to best fit their needs. FDMEE supports four types of scripts:

- Import scripts—Executed as source files are imported. Uses Jython script only.
- Mapping scripts—Used to build mapping logic to derive target members based on multiple source column and IF THEN logic. Uses Jython and SQL script.
- Event script—Executed in response to FDMEE such as before or after an import. Uses Jython and Visual Basic script.
- Custom script—Enables manual execution of scripts. Uses Jython and Visual Basic script.

FDMEE provides a set of Java APIs that enable you to look up frequently used data load properties or to perform a common action. FDMEE also supports Visual Basic based event scripts.

Key Jython Concepts

Before using scripts, be familiar with the following important Jython concepts:

- Code Indentation
- No Data Type Declaration
- Array Index starts at 0
- Substring is str[Start Position: Start Position + Length].
- ability to access any Java library

For information on Jython concepts: see:

- www.jython.org
- www.python.org

For information on using exception handling mechanisms in Jython, see www.jython.org.

For an example of how a "try-except block" is used in an Event script, see [Events Script Sample](#).

Using the Script Editor

The section explains how to use the Oracle Hyperion Financial Data Quality Management, Enterprise Edition Script Editor.

Overview

The Script Editor is used to define Jython scripts that run in response to Oracle Hyperion Financial Data Quality Management, Enterprise Edition events or during the file import processes. Scripts are saved in the data\scripts directory of the FDMEE application (with a .py extension for Jython scripts or a .vbs extension for Visual Basic scripts). Import scripts are saved to the data\scripts\import subdirectory, event scripts are saved to the data\scripts\event subdirectory, and custom scripts are saved to the data\scripts\custom subdirectory. Scripts can be copied to other FDMEE environments and edited using a text editor.

Launching the Script Editor

To access the Script Editor:

1. On the **Setup** tab, select **Scripts**.
2. Select **Script Editor**.

Script Editor Layout

The left pane of the Script Editor lists the directories that contain the two types of Oracle Hyperion Financial Data Quality Management, Enterprise Edition scripts: Import and Event. The right pane enables you to write and copy the code for script.

Using Import Scripts

This section explains how to use import scripts in Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

Overview

 Note:

Oracle Hyperion Financial Data Quality Management, Enterprise Edition import integration is implemented using the Jython scripting engine. The Visual Basic scripting engine cannot be used with import scripts.

When working with import scripts, note the following considerations:

- Import scripts are executed as source files are imported.
- Import scripts are only valid for file-based data loads.
- Import scripts must be created using Jython functions only. The format is "def <name>(parm1, parm2)" and a return value.
- Import script pass two parameters by FDMEE:
 - a dimension value defined by the import format
 - an entire record from the source

For example, when the account column is defined as position 2 to position 4 characters, then this value gets passed as the value for the field, and the whole input line is passed as the second parameter. In the sample script (see [Import Script Sample](#)) these parameters are specified by the StrField and StrRec values, but can be any names as long as you remember that the first parameter is the field, and the second one is the entire record.

- An import script on the Amount column is always executed first.
- The file name and the function name must be the same.
- The return value from the function populates the source column in the TDATASEG table for the specified dimension. You can only return a single value.
- It is recommended that you use a text editor to edit your script, and then copy and paste it into FDMEE.
- Import scripts are not handled in Oracle Hyperion Enterprise Performance Management System Lifecycle Management.

Creating Import Scripts

To create import scripts:

1. On the **Setup** tab, under **Scripts**, select **Script Editor**.
2. On the **Script Editor** screen, click **New**.
3. From **Script Type**, select **Import**.
4. In **File Name**, enter a name for the script.
5. Click **OK**.
6. Write the code for the custom script in the Script Editor.

7. Click **Save**.

Import Script Parameters

The import format script is defined as a Jython function. The Jython function name should be same as the script file name. This function accepts two parameters: the current Field and current row being processed.

- **strField**—The values of the source-file column that is defined in the import format (For example, for import scripts assigned to the Amount column, strField contains amount values, and, for import scripts assigned to the Account column, strField contains account values.
- **strRecord**—The source-file line (record) that is being scanned.

In following example, the script concatenates the account code with the account description from the source file. For example, the first data line of the file is account code 1110 with a description of “Cash”. This script concatenates 1110 with “Cash” and creates a source account in the TDATESEG table named “1110-Cash”. (See [TDATESEG Table Reference](#)). In the script, you select the field that is defined by the import format, and then the field defined by places 14 through 34 in the current record. (Jython starts numbering columns with 0 rather than 1.)

The screenshot shows the 'Import Format' dialog box with three main sections:

- Import Format Summary:** A table listing import formats. The first format, '0_MCPLAN1_IF', is selected.

Status	Name	Source System	Accounting Entity	Source Adapter
✓	0_MCPLAN1_IF	File		
✓	6_SALESJOURNAL	File		
✓	BRAZIL	File		
✓	CALIFORNIA	File		
- 0_MCPLAN1_IF: Details:** Configuration for the selected format.

Name	0_MCPLAN1_IF	Source System	File
* File Type	Fixed	File Delimiter	NA
Target Application	MCPLAN1	Drill URL	
* Request Method	GET	Description	
- 0_MCPLAN1_IF: Mappings:** A table mapping source columns to target fields.

Source Column	Start	Length	Expression	Add Expression	Target
Account	2	4			Account
Account3	47	14			Account

Assigning Import Scripts to Import Formats

After creating and saving a script in the Script Editor, you assign the script to an import field—any import field within any import format.

To assign import scripts to import fields:

1. On the **Setup** tab, under **Integration Setup**, select **Import Format**.
2. From the **Import Format summary grid**, select the import format name.



Note:

Use non-ASCII characters in an import format name when the import source is an adapter.

3. From the **Import Format detail grid**, select the **Add Expression** icon.
You can also type the value directly in the field rather than use the Expression editor.

Import Format [Save] [Cancel]

Import Format Summary

View [Add] [Delete] [Detach]

Status	Name	Source System	Accounting Entity	Source Adapter
✓	0_MCPLAN1_IF	File		
✓	6_SALESJOURNAL	File		
✓	BRAZIL	File		
✓	CAI IFORNTA	File		

0_MCPLAN1_IF: Details

Name: 0_MCPLAN1_IF Source System: File

* File Type: Fixed File Delimiter: NA

Target: MCPLAN1 Drill URL: []

Application: GET Description: []

* Request Method: GET

0_MCPLAN1_IF: Mappings

View [Detach] [Add] [Delete]

Source Column	Start	Length	Expression	Add Expression	Target
Account	2	4		[Add Expression]	Account
Account3	47	14		[Add Expression]	Account

4. From **Add Expression**, and then from **Expression Type**, select **Script**.
5. In **Expression Value**, browse to and select the script.

Add Expression [X]

* Expression Type: Script

* Expression Value: ModifySAPAmt.py

Description: Use this expression for using custom script during the data load process.

Example: Script=amount.py -> Uses script amount.py

[OK] [Cancel]

6. Click **OK**.

The script is assigned to the import field. The name of the import script is displayed in the Expression column.

Source Column	Start	Length	Expression	Add Expression	Target
Account	2	4	Script=ModifySAPAmt.py		Account
Account3	47	14			Account
Amount	114	18			Amount

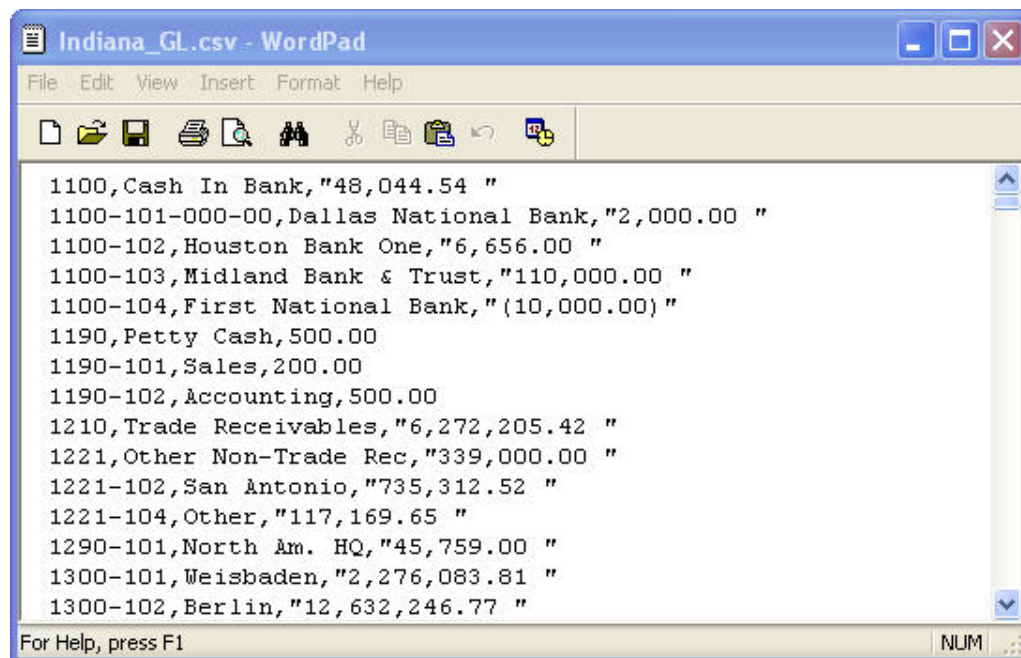
Using Import Script Functions

Within import scripts, you use functions to manipulate the data that Oracle Hyperion Financial Data Quality Management, Enterprise Edition imports.

This section uses a problem/solution format to explain how to use the various import script functions.

Extracting Beginning Characters

Problem: The account numbers of the Indiana_GL screen, which are in the first field of each line, include varying numbers of digits. However, the mapping associated with the file uses only four digits.



Solution: In the Import Scripts screen, assign the following script to the Expression field of the Account row. The script assigns the first four-digits of each account number (the first four digits on the left) to Parse_Account:

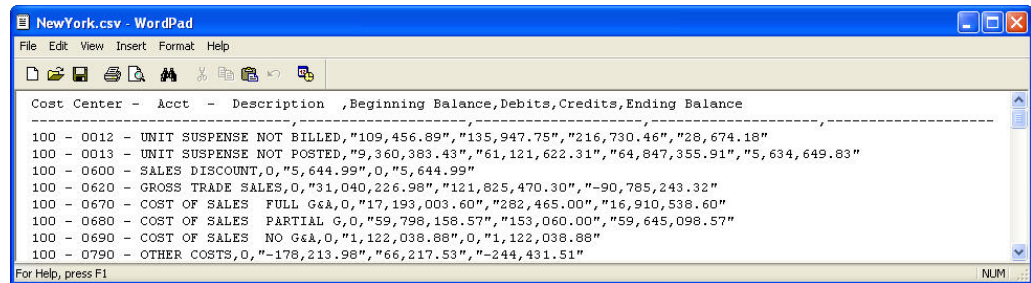
```
def Parse_Account (strfield, strrecord):

    return strField[:4]
```

Result: The import file displays only the first four digits of each account number.

Extracting Middle Characters

Problem: The NewYork file presents cost center, account, and account description as a continuous string, rather than as three strings. You want to separate the account strings from the cost center and account description strings.



Solution: In the Import Formats screen, assign the following script to the Expression field of the Account row. In the Import Formats screen, assign the following script to the Expression field of the Account row. The script extracts and returns the characters of the account strings (the characters from positions 7 to 10 of the string):

```
def Parse_Account (strfield, strrecord):

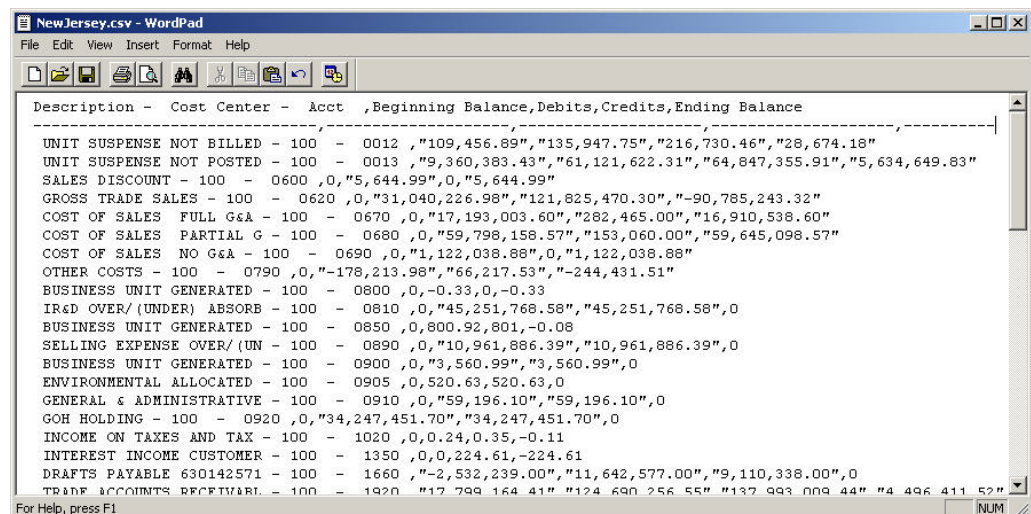
    return strField[6:10]
```

Result: Account strings are separated from cost center and account description strings.

Result: In the Account column of the import file, only account numbers are displayed.

Extracting End of String

Problem: The NewJersey screen presents account numbers as the last four characters of account fields. You want to extract only account numbers.



Solution: In the Import Formats screen, assign the following script to the Expression field of the Account row. The script extracts and returns the account numbers (the last four characters of the account fields):

```
def Parse_Account (strfield, strrecord):

return strField[-4:]
```

Result: In the Account column of the import file, only account numbers are displayed.

Using Split Functions

Problem: The NewYork screen presents cost center, account, and account description in one field, rather than in three fields. The three strings are separated by dashes (-). You want to present the three strings as three fields.

Cost Center	Acct	Description	Beginning Balance	Debits	Credits	Ending Balance
100 - 0012	-	UNIT SUSPENSE NOT BILLED,	'109,456.89'	'135,947.75'	'216,730.46'	'28,674.18'
100 - 0013	-	UNIT SUSPENSE NOT POSTED,	'9,360,383.43'	'61,121,622.31'	'64,847,355.91'	'5,634,649.83'
100 - 0600	-	SALES DISCOUNT,0,"	'5,644.99',0,"	'5,644.99'		
100 - 0620	-	GROSS TRADE SALES,0,"	'31,040,226.98',	'121,825,470.30',	'-90,785,243.32'	
100 - 0670	-	COST OF SALES FULL G&A,0,"	'17,193,003.60',	'282,465.00',	'16,910,538.60'	
100 - 0680	-	COST OF SALES PARTIAL G,0,"	'59,798,158.57',	'153,060.00',	'59,645,098.57'	
100 - 0690	-	COST OF SALES NO G&A,0,"	'1,122,038.88',	'0',	'1,122,038.88'	
100 - 0790	-	OTHER COSTS,0,"	'-178,213.98',	'66,217.53',	'-244,431.51'	

Solution: In the Import Formats screen, assign the following scripts, each of which uses the split function, to the Expression fields of the Entity, Account, and Account Description rows (first script to Entity, second script to Account, and third script to Account Description). The first script returns the set of characters before the first hyphen (a cost center value), the second script returns the set of characters after the first hyphen (an account value), and the third script returns the set of characters after the second hyphen (an account description value).

```
def NY_ParseCenter (strfield, strrecord):

    seglist = strfield.split("-")

    return seglist[0].strip()

def NY_ParseAccount (strfield, strrecord):

    seglist = strfield.split("-")

    return seglist[1].strip()

def NY_ParseDesc (strfield, strrecord):

    seglist = strfield.split("-")

    return seglist[2].strip()
```

Result: In the import file, cost center, account and account description strings are presented in three separate fields.

Using the Skip Function (Conditional Skip)

Problem: You want Oracle Hyperion Financial Data Quality Management, Enterprise Edition to skip all lines of the NewYork screen that contain an entity value that begins with 06.

You must specify `fdmSkip` as the return value from your function in order to skip a line in the file that is being processed in your import script. You specify `fdmSkip` as the return argument from an import script by entering `return fdmSkip`. `fdmSkip` is only available for import scripts.

Cost Center	Acct	Description	Beginning Balance	Debits	Credits	Ending Balance
100 - 0012	-	UNIT SUSPENSE NOT BILLED	"109,456.89"	"135,947.75"	"216,730.46"	"28,674.18"
100 - 0013	-	UNIT SUSPENSE NOT POSTED	"9,360,383.43"	"61,121,622.31"	"64,847,355.91"	"5,634,649.83"
100 - 0600	-	SALES DISCOUNT	0	"5,644.99"	0	"5,644.99"
100 - 0620	-	GROSS TRADE SALES	0	"31,040,226.98"	"121,825,470.30"	"-90,785,243.32"
100 - 0670	-	COST OF SALES FULL G&A	0	"17,193,003.60"	"282,465.00"	"16,910,538.60"
100 - 0680	-	COST OF SALES PARTIAL G	0	"59,798,158.57"	"153,060.00"	"59,645,098.57"
100 - 0690	-	COST OF SALES NO G&A	0	"1,122,038.88"	0	"1,122,038.88"
100 - 0790	-	OTHER COSTS	0	"-178,213.98"	"66,217.53"	"-244,431.51"

Solution: In the Import Scripts screen, you assign the following script to the Expression field of the Entity row. The script parses the entity column and uses a local variable that is discarded after the script executes:

```
def NY_Skip06Center(strField, strrecord):

    if strfield.count("-") > 0:

        seglist = split(strField, "-")

        strEntity = seglist[0]

        if strEntity[:2] == "06":

            return fdmSkip

        else:

            return strEntity
```

Result: No line that contains entity values that begin with 06 is imported.

Storing and Retrieving Temporary Variables

When Oracle Hyperion Financial Data Quality Management, Enterprise Edition imports a source file, it skips lines that do not contain valid amounts but executes all import scripts assigned to the Amount column first regardless of whether amounts are valid. Therefore, you can use scripts that run for lines that FDMEE would otherwise skip to store global variables that can be retrieved by other scripts.

Storing Temporary Variables

Within source files, not all lines contain all fields. For example, in the Georgia screen shown below, entity values, such as 0565 0565 Test 3, are presented in the header of

each report section after the Bus Area/Dept label, but are not presented in every line. Therefore, entity values must be stored in global variables and assigned to the lines of the source file.

For the Georgia file, to store entity values in global variables, in the Import Formats screen, you assign the following script to the Expression field of the Amount row. The script uses an `if..` statement and the string function to determine whether lines contain the Bus Area / Dept: label. If a line contains the label, the script stores the entity value (located in position 33 and including 4 characters) in a global variable. If a line does not include the label, `strfield` returned.

To use a global variable, define a string variable outside the import function. When assigning a value to the variable inside the import function, designate it as global. By defining a variable outside the function, it is available to any function used in that session.

The global variables remain available throughout the current Oracle Hyperion Financial Data Quality Management, Enterprise Edition data load process (unlike local variables, which lose their values when the current script is closed). Therefore, values stored from within one script can be retrieved by other scripts.

```
GeorgiaCenter = ""
```

```
def GeorgiaGetCenter (strfield, strrecord):

    if strrecord[15:31] == "Bus Area / Dept:":

        global GeorgiaCenter

        GeorgiaCenter = strrecord[32:36]

    return strfield
```

Georgia.glo - Notepad

Upstream Software Summary1 Trial Balance Report Date: 16-DEC-2003 13:08
Period: NOV03-04 Page: 44 of 63

Currency: USD
Balance Type: Year to Date
Bus Area / Dept Range: 0000 to 0999
Bus Area / Dept: 0563 0563 - Test1

Acct	Description	Beginning Balance	Debits	Credits	Ending Balance
0010	0010 - Cash	(971,295.74)	951.00	0.00	(970,344.74)
0012	0012 - AP	0.00	2,002.00	2,002.00	0.00
		(971,295.74)	2,953.00	2,002.00	(970,344.74)

Upstream Software Summary1 Trial Balance Report Date: 16-DEC-2003 13:08
Period: NOV03-04 Page: 45 of 63

Currency: USD
Balance Type: Year to Date
Bus Area / Dept Range: 0000 to 0999
Bus Area / Dept: 0564 0564 - Test2

Acct	Description	Beginning Balance	Debits	Credits	Ending Balance
0010	0010 - Cash	5,666,682.18	1,282,699.97	6,949,282.15	100.00
0012	0012 - AP	0.00	403.00	403.00	0.00
		5,666,682.18	1,283,102.97	6,949,785.15	0.00

Upstream Software Summary1 Trial Balance Report Date: 16-DEC-2003 13:08
Period: NOV03-04 Page: 46 of 63

Currency: USD
Balance Type: Year to Date
Bus Area / Dept Range: 0000 to 0999
Bus Area / Dept: 0565 0565 - Test3

Acct	Description	Beginning Balance	Debits	Credits	Ending Balance
0010	0010 - Cash	44,521,731.20	165,879,142.19	269,201,268.90	(58,800,395.51)

Retrieving Temporary Variables

You use scripts to assign global, stored variables to the fields of a column. For example, if you are working with the Georgia screen, you begin by using the following script to assign the global variable `GeorgiaCenter` to the **GeorgiaPutCenter** function.

```
def GeorgiaPutCenter (strfield, strrecord)

return GeorgiaCenter
```

Then, in the Import Formats screen, you assign the script to the Expression field of the Entity row and, thus, assign the values of the global variables to the Entity fields. In this case, entity values are not read from the source file.

Because the Georgia file includes subtotal lines that must not be imported, the skip function is required.

To direct Oracle Hyperion Financial Data Quality Management, Enterprise Edition to skip lines without account numbers, you configure a Skip Row in the Import Format to recognize blank Account fields (15 blank spaces) by defining appropriate start and length fields for the expression.

Georgia.glo - Notepad

File Edit Format View Help

Upstream Software Summary1 Trial Balance Report Date: 16-DEC-2003 13:08
Period: NOV03-04 Page: 44 of 63

Currency: USD
Balance Type: Year to Date
Bus Area / Dept Range: 0000 to 0999
Bus Area / Dept: 0563 0563 - Test1

Acct	Description	Beginning Balance	Debits	Credits	Ending Balance
0010	0010 - Cash	(971,295.74)	951.00	0.00	(970,344.74)
0012	0012 - AP	0.00	2,002.00	2,002.00	0.00
		(971,295.74)	2,953.00	2,002.00	(970,344.74)

Upstream Software Summary1 Trial Balance Report Date: 16-DEC-2003 13:08
Period: NOV03-04 Page: 45 of 63

Currency: USD
Balance Type: Year to Date

Import Script Sample

This sample import script returns the location as a value for the column.

```
#-----
# Sample shows how to use the value from the fdmContext map, In
# this sample return the Location as value for the column
#-----

def getOrgfromLoc(strfield, strrec):

    org = fdmContext['LOCNAME']

    return org
```

```
#-----  
# Sample to show the Jython string function. The script below is  
# used to parse an account column 01-205-4110-0000-000 and return the  
# third string  
#-----  
  
def getSegfromAcct(strfield, strrec):  
    if strfield.count("-") > 0:  
        seglist = strfield.split('-')  
        result = seglist[2]  
        return result  
#-----  
  
# Sample to process header record in Trial Balance report and  
# store value in global variable to be used during detail row  
# This should be called from the Amount column in import format  
#-----  
  
globalorg = ""  
globalcur = ""  
  
def copyglobal(strfield, strrec):  
    if strrec[18:27] == "Currency:" :  
        global globalcur  
        globalcur = strrec[29:32]  
    if strrec[14:27] == "Organization:" :  
        global globalorg  
        globalorg = strrec[29:32]  
    if strrec[42:47].strip() == 'Total' or strrec[29:34].strip() == 'Total' :  
        return fdmSkip  
    return strfield  
#-----  
  
# Sample to return the global variable set during amount process  
#-----  
  
def getCurrfromGlobal(strfield, strrec) :
```

```
return globalcur
```

Using Mapping Scripts

This section explains how to use mapping scripts in Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

Overview

Mapping Scripts are used to build mapping logic to derive target members based on multiple source columns and IF THEN logic. Mapping scripts are added in the Data Load Mapping screen, and are available for: Between, IN, Like types. They are stored in the TDATAMAP table.

When you use Oracle Hyperion Enterprise Performance Management System Lifecycle Management to export mapping rules, any related mapping scripts are included.

Additionally Oracle Hyperion Financial Data Quality Management, Enterprise Edition supports the export and import of mapping scripts in a text file. This support includes both Jython and SQL scripts. The scripts are enclosed in a <!SCRIPT> tag.

Creating Mapping Scripts

For Like mappings, you can create mapping scripts to designate conditional mapping. This type of mapping enables you to specify source members mapped to script expressions rather than to hard-coded target members. Target values are assigned in the script value. You activate the mapping script by placing #SQL in the Target value field for a SQL script. (The row that uses a script has target value of #SQL). Then the next row has the <!SCRIPT> delimiter and the actual script, followed by <!SCRIPT>. Mapping scripts, used in conjunction with dimension processing order, enables a mapping that is based on the results of dimension mappings. That is, dimension mappings that have already been processed. See [Using Mapping Scripts](#).

To create a mapping script:

1. From the **Workflow** tab, select **Data Load Mapping**.
2. **Optional:** Select the desired location.
3. Select the **Like** tab.

Mapping scripts are unavailable for "Explicit" and "Multi-dimension" mapping types.
4. Select the **Source Value**.
5. In **Target Value**, select one of the following:
 - For a Jython based mapping script, enter **#SCRIPT**.
 - For a SQL based mapping script, enter **#SQL**.

 Note:

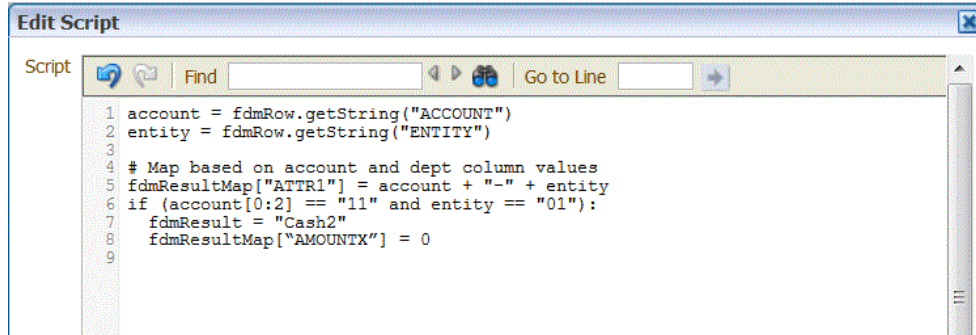
For SQL based mappings, Oracle Hyperion Financial Data Quality Management, Enterprise Edition specifies the special value #SQL to the "Target Value." For this reason, this field cannot be used to specify a value of "IGNORE", which is otherwise used to set the VALID_FLAG to Yes or No. For SQL based mappings, you must include the VALID_FLAG in the SQL script and specify the value of "I" based on the user selected criteria.

6. In **Rule Name**, enter the data rule name for the mapping.
7. Click **Save**.

The Script Editor icon () is enabled.

8. Click the Script Editor icon.
9. In **Edit Script**, enter the mapping script, and click **OK**.

For example the following Jython based mapping script checks when the first two characters of the account equals 11 and the entity equals 01. When a match occurs, then the target is set to Cash2 and the target amount is set to 0. In addition, the script also updates the attribute1 column in the TDATESEG table. (See [TDATESEG Table Reference](#)). This script uses the fdmResultMap object (see [Using Jython Objects with Mapping Scripts](#)).

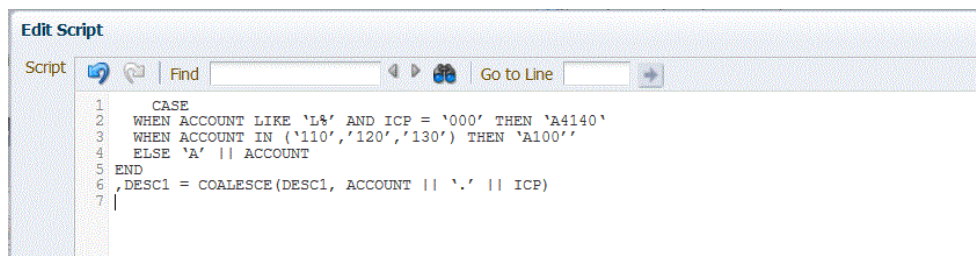


```

1 account = fdmRow.getString("ACCOUNT")
2 entity = fdmRow.getString("ENTITY")
3
4 # Map based on account and dept column values
5 fdmResultMap["ATTR1"] = account + "-" + entity
6 if (account[0:2] == "11" and entity == "01"):
7     fdmResult = "Cash2"
8     fdmResultMap["AMOUNTX"] = 0
9

```

In this example, SQL script is used logic is implemented in a SQL CASE statement. The CASE statement is used in the SET command of the internal UPDATE statement. The mapping statement below is converted to the UPDATE statement listed



```

1 CASE
2   WHEN ACCOUNT LIKE 'L%' AND ICP = '000' THEN 'A4140'
3   WHEN ACCOUNT IN ('110','120','130') THEN 'A100'
4   ELSE 'A' || ACCOUNT
5 END
6 ,DESC1 = COALESCE(DESC1, ACCOUNT || '.' || ICP)
7

```

```

UPDATE TDATESEG

SET ACCOUNTX =

CASE

    WHEN ACCOUNT LIKE 'L%' AND ICP = '000' THEN 'A4140'

    WHEN ACCOUNT IN ('110','120','130') THEN 'A100'

    ELSE 'A' || ACCOUNT

END

,DESC1 = COALESCE(DESC1, ACCOUNT || '.' || ICP)

WHERE ACCOUNT ....

```

Using Jython Objects with Mapping Scripts

Use the following predefined Oracle Hyperion Financial Data Quality Management, Enterprise Edition Jython objects within mapping scripts.

Table 7-1 Using Jython Objects with mapping scripts

Jython Object	Description
fdmRow	fdmRow id used to access any column in the current row being processed by the mapping process. You can access any column in the TDATESEG table. To access a value in a column, specify the following: fdmRow.getString("<COLUMN NAME>"). For example, to get the value of the ENTITY column, specify fdmRow.getString("ENTITY").
fdmResult	fdmResult is used to return the value from the mapping function. You can set the value as follows fdmResult = "Cash". The fdmResult is used to update the value for the target dimension, which is also referred to as the "X" column. For every dimension in the TDATESEG table there are two columns, one for source and another for target. For example, ENTITY provides the source value, and ENTITYX provides target value that is the result of the mapping process.
fdmResultMap	fdmResultMap is used to update any column in the TDATESEG table. The column name in the TDATESEG table is used as the key in the map. For example, to update the ATTR1 column use fdmResultMap["ATTR1"] = "Asset Acct". To set the target amount to 0 use fdmResultMap["AMOUNTX"] = 0.

Mapping Script Samples

This sample mapping script evaluates the account and entity columns and assigns a value for the target column. It also shows how to update the other columns of the current row using the `fdmResult` map:

```
#-----  
  
# Sample Jython Mapping Script. Script evaluates account and entity  
# columns and assign value for the target column. In addition it  
# also shows how to update the other columns of current row using  
# fdmResultMap  
#-----  
  
account = fdmRow.getString("ACCOUNT")  
entity = fdmRow.getString("UD1")  
  
# Map based on account and dept column values  
fdmResultMap["ATTR1"] = account + "-" + entity  
  
if (account[0:2] == "71"):  
    fdmResult = "7110"  
  
elif (account[0:2] == "72"):  
    fdmResult = "7210"  
  
elif (account[0:2] == "77" and entity == "205"):  
    fdmResult = "7710"  
  
    fdmResultMap["AMOUNTX"] = 0  
  
else:  
    fdmResult = "7310"
```

This sample script uses the SQL CASE statement to conditionally process assigned values for the target column.

```
#-----  
  
# Sample SQL script. Script uses SQL CASE statement to conditionally  
# process assign value for the target column.  
#-----  
  
CASE  
  
    WHEN ACCOUNT LIKE '61%'    AND ud1 = '205'    THEN '6110'
```

```
WHEN ACCOUNT LIKE '61%' AND ud1 = '240' THEN '6120'

ELSE '6130'

END
```

Using Event Scripts

This section explains how to use Event scripts in Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

Overview

Event scripts are executed in response to Oracle Hyperion Financial Data Quality Management, Enterprise Edition events. You can invoke any custom logic. For example, custom logic could be invoked to send an email notification after a successful load, or send an email when a validation error occurs. Or an email could be sent when you download data from Oracle Hyperion Financial Management and load data to Oracle Essbase for reporting. Event scripts are based on Jython or Visual Basic. Jython scripts have a .py extension, and Visual Basic scripts have a .vbs extension. Events associated with a script are highlighted in the list of events. Any event that includes a script is executed for that event in selected location.

Note:

Event scripts are not handled in Oracle Hyperion Enterprise Performance Management System Lifecycle Management.

FDMEET Supported Event Scripts

Oracle Hyperion Financial Data Quality Management, Enterprise Edition supports the following events for execution, during the data load process:

Table 7-2 FDMEET Events

Event	Description
BefImport	This event is the initial state of the system before any processing for the selected location has begun. If the user defines import scripts, they are run between the BefImport and AftImport events. This step in the processing loads data into the TDATESEG_T table.
AftImport	Data is present in the TDATESEG_T table after this event is processed.
BefCalculate	Called for a validation run only, and called before the validation process.
AftCalculate	Called for a validation run only, and called after the validation process.
BefProcLogicGrp	Called before Logic Account is processed.
AftProcLogicGrp	Called after the Logic Account is processed.

Table 7-2 (Cont.) FDMEE Events

Event	Description
BefProcMap	Called before the mapping process is started in the TDATESEG_T table. Mapping scripts are executed between the BefProcMap and AftProcMap events. Data is moved from the TDATESEG_T table to the TDATESEG table between these events after all data has been mapped. Updates to audit tables are also included as part of this process.
AftProcMap	Called after all the mapping has been completed. When this event runs, the data has already been moved from the TDATESEG_T table to the TDATESEG table.
BefValidate	Checks if data mapped in the TDATESEG table.
AftValidate	Called after the BefValidate event.
BefExportToDat	Called before you write to the file for export. It is also executed for Oracle Essbase in the export to file mode feature.

**Note:**

This event is unavailable for the Accounts Reconciliation Manager.

AftExportToDat

Called after the file is created.

**Note:**

This event is unavailable for the Accounts Reconciliation Manager.

Table 7-2 (Cont.) FDMEE Events

Event	Description
BefLoad	Called before you load to a target application.
	 Note: This event is unavailable for the Accounts Reconciliation Manager.
AftLoad	Called after data is sent to the target application, but does not wait for the target application to complete the load process.
	 Note: This event is unavailable for the Accounts Reconciliation Manager.
BefConsolidate	Oracle Hyperion Financial Management and Essbase only: This event executed when a check rule is included in the location that is being processed.
	 Note: This event is unavailable for the Accounts Reconciliation Manager.

Table 7-2 (Cont.) FDMEE Events

Event	Description
AftConsolidate	Called after the BefConsolidate event.
	 Note: This event is unavailable for the Accounts Reconciliation Manager.
BefCheck	Called before the Check Rule.
	 Note: This event is unavailable for the Accounts Reconciliation Manager.
AftCheck	Called after the Check Rule.
	 Note: This event is unavailable for the Accounts Reconciliation Manager.

Creating Event Scripts

To create event scripts:

1. On the **Setup** tab, under **Scripts**, select **Script Editor**.
2. Single or double-click an event.
3. Write the script in the Script Editor.

 Note:

Before creating event scripts, make sure you check the Application Root Folder setting in Application Settings. When you specify a folder at the application level, select **Create Application Folders**. A set of folders is created for the application that includes a scripts folder. Create scripts specific to an application in this folder. This is especially important for event scripts that are different between applications. When you do not set up an application level folder, then you cannot have different event scripts by application.

Stopping the Execution of FDMEE from within a Script

To halt the execution of Oracle Hyperion Financial Data Quality Management, Enterprise Edition from within a script:

1. On the **Setup** tab, under **Scripts**, select **Script Editor**.
2. Single or double-click an event.
3. Add the **raise RuntimeError** script and the message to display.

For example,

```
if (categoryName == ""):

    raise RuntimeError, "Category name is invalid"
```

4. Save the script.

Dynamically Changing Import Formats

You can use the `BefFileImport` event to modify the import format for a location dynamically. The following script changes the import group; the change is based on the file name.

To implement the script, you create two import formats and assign one of them to the Oracle Hyperion Financial Data Quality Management, Enterprise Edition location. The script evaluates the file name and, if necessary, changes the import format.

```
if fdmContext["LOCNAME"] == "ITALY":

    filename = fdmContext["FILENAME"]

    if filename[:12] == "ProductSales":

        fdmAPI.updateImportFormat("SALESJOURNAL", fdmContext["LOADID"])
```

Using the File System Object in Event Scripts

You can use the Jython file system object to process files and folders. The following example uses the file system object to create a file and to copy the contents of an existing file to the new file.

Read the following Input File

```
Entity,Currency,ICP,Product,Store,Channel,Custom4,Custom5,Custom6,Custom7,UnitsSold,Sales
```

```

EastSales, USD, [ICP None], H740, Comma_Phone_Stores, Retail_Direct, [None],[None],
[None],[None],127,9954.103768

EastSales, USD, [ICP None], H740, Freds, National_Accts, [None],[None],[None],[None],
112,6610.371552

EastSales, USD, [ICP None], H740, Good_Buy, National_Accts, [None],[None],[None],
[None],112,6610.371552

Write the following Output File

EastSales, USD, [ICP None], H740, Comma_Phone_Stores, Retail_Direct, [None],[None],
[None],[None],UnitsSold,127

EastSales, USD, [ICP None], H740, Comma_Phone_Stores, Retail_Direct, [None],[None],
[None],[None],Sales,9954.103768

EastSales, USD, [ICP None], H740, Freds, National_Accts, [None],[None],[None],
[None],UnitsSold112

EastSales, USD, [ICP None], H740, Freds, National_Accts, [None],[None],[None],
[None],Sales6610.371552

EastSales, USD, [ICP None], H740, Good_Buy, National_Accts, [None],[None],[None],
[None],UnitsSold,112

EastSales, USD, [ICP None], H740, Good_Buy, National_Accts, [None],[None],[None],
[None],Sales,6610.371552

infilename = fdmContext["INBOXDIR"]+"/InputFile.txt"

outfilename = fdmContext["INBOXDIR"]+"/DataFile.txt"

infile = open(infilename, "r")

outfile = open(outfilename, "w")

for line in infile:

    column = line.rsplit(',',2)

    if column[2].strip() != "Sales" :

        outfile.write(column[0] + ",UnitsSold," + column[1] + "\n")

        outfile.write(column[0] + ",Sales," + column[2])

outfile.close()

```

Events Script Sample

This sample Event script updates the table_xyz table during the data load execution:

```

#-----
# Sample to update table_xyz table during data load rule execution
#-----

```

```
query = "UPDATE table_xyz SET accountx = 'SCRIPT_' || account WHERE loadid = ? and
accountx is NULL"

params = [ fdmContext["LOADID"] ]

print fdmAPI.executedDML(query, params, False)

fdmAPI.commitTransaction()

#-----

# Sample to import data from a custom SQL source and upload into FDMEE
# open interface table. This script should be called in BefImport Event.
# This is alternate to the FDMEE integration import script.
#-----

import java.sql as sql

batchName = "Batch_" + str(fdmContext["LOCNAME"])

insertStmt = ""

INSERT INTO AIF_OPEN_INTERFACE (

    BATCH_NAME

    ,COL01

    ,COL02

    ,AMOUNT

    ,DESC1

) VALUES (

?

,?

,?

,?

,?

)

""

sourceConn = sql.DriverManager.getConnection("jdbcUrl", "user", "password");

# Limiting number of rows to 5 during the test runs.

selectStmt = "SELECT * FROM orders WHERE rownum < 6"

stmt = sourceConn.prepareStatement(selectStmt)
```

```
stmtRS = stmt.executeQuery()
while(stmtRS.next()):
    params = [ batchName, stmtRS.getBigDecimal("Customer_Id"),
               stmtRS.getString("Ship_Country"),
               stmtRS.getBigDecimal("Freight"), stmtRS.getString("Ship_Name") ]
    fdmAPI.executedDML(insertStmt, params, False)
fdmAPI.commitTransaction()
stmtRS.close()
stmt.close()
sourceConn.close()

#-----
# Sample to send email messages using Jython SMTP library
#-----

import smtplib
sender = "from@gmail.com"
receivers = "to@gmail.com"
message = """ This is a test e-mail message.
               This is a test e-mail message. """

try:
    smtpServer = smtplib.SMTP('smtp.gmail.com:587')
    smtpServer.starttls()
    smtpServer.login("user", "password")
    smtpServer.sendmail(sender, receivers, message)
    print "Successfully sent email"
    smtpServer.quit()
except Exception, e:
    print "Error: unable to send email: " + str(e)
```

 Note:

See the online Jython documentation at the following link that explains the list of Jython exceptions, and the syntax to use when trapping exceptions in your scripts: [The Definitive Guide to Jython](#)

This note applies to all script types.

Using Custom Scripts

This section shows how to use custom scripts in Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

Overview

Oracle Hyperion Financial Data Quality Management, Enterprise Edition enables you to perform FDMEE tasks on an "as needed" basis such as executing data load rules using custom scripts.

FDMEE supports custom scripts in Jython and Visual Basic. To execute a custom ODI process, then use a Jython script. FDMEE stores custom scripts in the <APP DIR>/data/scripts/custom folder.

Creating a Custom Script

To create a custom script:

1. On the **Setup** tab, under **Scripts**, select **Script Editor**.
2. On the **Script Editor** screen, click **New**.
3. From **Script Type**, select **Custom**.
4. From **Technology**, select **Jython** or **Visual Basic**.

Scripts created in Jython are saved with a .py extension. Scripts created in Visual Basic are saved with a .vbs extension.

5. In **File Name**, enter a name for the script.
6. Click **OK**.
7. Write the code for the custom script in the Script Editor.
8. Click **Save**.

Working with Custom Scripts

You must create custom script groups before registering custom scripts.

You register custom scripts to select the parameters to pass when the script is executed.

Adding a Custom Script Group

Custom scripts groups enable you to assign similar types of custom scripts under a group for ease of use. They are also used to determine security eligibility when executing custom scripts.

To add a custom group:

1. On the **Setup** tab, under **Scripts**, select **Script Registration**.
2. Select the **Custom Script Group** tab.
3. Click **Add**.
4. In **Name**, specify the custom script group name.
5. In **Sequence**, enter the display order used when custom script groups are shown during registration and execution.
6. Click **Save**.

Registering Scripts

Registered scripts are associated with a script file (which consists of Jython or Visual Basic script) and a target application. When the custom script is executed, you are prompted with a list of values from which to complete the parameters of the script.

The screenshot shows the 'Script Registration' window. It has a left sidebar with 'Setup' and 'Scripts' tabs. The 'Scripts' tab is active, showing 'Script Registration'. The main area is divided into three sections: 'Summary', 'Details', and 'Parameters'.

Name	Custom Script Group	Script File	Target Application
KSSCRPT	KB Script Group	kscript.py	COMMA7
KB_SCRIPT_2	CSG2	aaaa.vbs	KBHFM2
KB_SCRIPT_1	CSG1	TestScript.vbs	Custom App 1
TestParamReg1	CSG1	TestParams.vbs	COMMA4
KB_LCM_SCRIPT_1	KB_LCM_GRP	KScript1.py	KB_LCM_TEST

Details

Name: KSSCRPT
 Custom Script Group: KB Script Group
 Target Application: COMMA7
 Script File: kscript.py

Script Parameters

Parameter Name	Type	Display Name	Sequence	Parameter Value	Query
LOCHNAME	POV Location	Location	1		

To register a script:

1. On the **Setup** tab, under **Scripts**, select **Script Registration**.
 The Custom Script Registration screen consists of three regions:
 - **Summary**—Lists all custom scripts.
 - **Details**—Shows the script name, associated target application, and script file.
 - **Parameters**—Shows the parameter name and type, display name, sequence, parameter value, and any query definition used to supply a list of values for a given custom script.
2. Above the Summary grid, click **Add**.
3. In the Details grid, in **Name**, enter the name of the custom script.

4. In **Target Application**, select the target application associated with the custom script.
5. To associate the custom script with a custom script group, in **Custom Script Group**, select the group.
6. In **Script File**, select the script file to associate with the custom script.

To search on a script file, click **Select** and choose a script file from the **Select Script File** screen.

To upload a script file, click **Select**. On the **Select Script File**, click **Upload**. Then click **Browse** to navigate to the script file to upload and click **OK**.

7. Click **Save**.

To define the parameters of the custom script:

1. From the Summary grid, select the name of the custom script to which to add parameters.
2. In the **Script Parameters** grid, click **Add**.

A blank line is displayed to which to add parameters.

3. In **Parameter Name**, enter the name of the parameter that you reference in the script.

For example, enter `POVL0C`, for the POV location, `POVPeriod` for the POV period, `POVCAT` for POV category, or `POVTARGET` for the target category.

The parameter name is user-defined. You can select a prefix or suffix to distinguish them in the custom script.

4. In **Type**, select the type of parameter.

Available types:

- **POV**—Prompts for the POV location, period, category or rule. The POV values default from the users' POV when the custom script is executed.
- **Query**—The Query type enables you to create a query that populates a list of values that a user can select from when the custom script is executed.
- **Static**—A Static type indicates that the parameter is a constant or a predefined value, and the parameter value is supplied by the user. When the custom script is executed, you can supply a different value to the parameter.

You can use any name for the parameter and also use a prefix or suffix with the name to distinguish them in the custom script.

The parameter name must exactly match what is provided in the script registration

5. In **Display Name**, enter the name to display for the parameter for the Execute Script screen.
6. In **Parameter Value**, enter the value for the "Static" parameter type.

The parameter value must be retrieved in the custom script using the following API:

```
fdmAPI.getCustomScriptParameterValue("PARAMETER NAME"))
```

7. In **Sequence**, enter the display order of the parameter on the Execute Script screen.

8. In **Query**, specify the query definition that provides a list of values for the parameter.
For example, to show a list of data rules for the user to select on the Generate Report screen, specify the query definition to define these queries.
9. Click **Save**.

Executing a Custom Script

Custom scripts can be executed using the Script Execution option, or from a command line.

To execute a custom script from the Script Execution option:

1. On the **Workflow** tab, under **Other**, select **Script Execution**.
2. In **Script Execution**, and then in **Custom Script Group**, select a custom script group.
3. From the **Scripts** grid, select the custom script.
4. Click **Execute**.
5. When prompted, enter parameter values on the Execute Script screen.
 - a. If applicable, modify the **Period**, **Category**, and **Location** values.
 - b. From **Execution Mode**, select the online method of running the report.
The online method processes the report immediately.
 - c. Click **OK**.

Executing a Custom Script from a Command Line

Oracle Hyperion Financial Data Quality Management, Enterprise Edition allows you to execute custom scripts from a command line. You can run the custom script with or without parameters. If you run a script that has no parameter, FDMEE determines the application id associated with the script. If you want to pass parameters, add parameters using the format: Executescript <script name> <Parameter Display Name>=<Value>.

1. At a command line, type the name of custom script to execute.
2. Click **Enter** to execute the script.

Custom Script Sample using Jython

These sample custom scripts provide information about the contents of the custom script.

The first example shows how to submit a data load rule by way of a script.

The second example shows how to submit a report using a batch file by way of a script.

Submitting a Data Load Rule

This example shows how to submit a data load rule by way of a script.

```

#

This sample Jython script provides code related to custom scripts. All the messages
being logged (meaning printed) are visible in the process lLog Level profile.

#

# Log the beginning of the script, at the INFO level

fdmAPI.logInfo("=====
=)

fdmAPI.logInfo("Custom Script: Begin")

fdmAPI.logInfo("=====
=)

# Log the target application name from the context, at the DEBUG level

fdmAPI.logDebug("Target App Name    = " + fdmContext["TARGETAPPNAME"])

# Log the script name from the context at the DEBUG level

fdmAPI.logDebug("Custom Script name = " + fdmContext["SCRIPTFILE"])

# Get all script parameters and log their names and values at the DEBUG level

fdmAPI.logDebug("Custom script parameter values by name: begin")

fdmAPI.logDebug("The value of parameter CUSTOM_LOCATION is = " +
fdmAPI.getCustomScriptParameterValue("CUSTOM_LOCATION"))

fdmAPI.logDebug("The value of parameter CUSTOM_YEAR is = " +
fdmAPI.getCustomScriptParameterValue("CUSTOM_YEAR"))

fdmAPI.logDebug("Custom script parameter values by name: end")

# Example submitting a data rule

import os

import subprocess

os.chdir("C:/Oracle/Middleware/user_projects/epmsystem1/FinancialDataQuality")

myScriptName = "C:/Oracle/Middleware/user_projects/epmsystem1/FinancialDataQuality/
loaddata.bat"

command = '%s "%s" "%s" "%s" "%s" "%s" "%s" "%s" "%s" "%s" "%s" "%s"' %
(myScriptName, "admin", "password", "SRESBA1_DR1", "Y", "N", "STORE_DATA",
"SNAPSHOT", "N", "Jan-2003", "Jan-2003", "ASYNC")

fdmAPI.logDebug("Submitting a data rule via a script using the following command: "
+ command)

retcode = subprocess.Popen(command)

fdmAPI.logDebug("Data rule submitted fine.")

# Close the connection and log the end of the script, at INFO level

```

```

fdmAPI.closeConnection()

fdmAPI.logInfo("=====
=)

fdmAPI.logInfo("Custom Script: end")

fdmAPI.logInfo("=====
=)

```

The output for the executed custom script is:

```

2013-09-25 08:12:26,080 INFO [AIF]:
=====

2013-09-25 08:12:26,081 INFO [AIF]: Custom Script: Begin

2013-09-25 08:12:26,082 INFO [AIF]:
=====

2013-09-25 08:12:26,083 DEBUG [AIF]: Target App Name = SRESBA1

2013-09-25 08:12:26,084 DEBUG [AIF]: Custom Script name = SRCustomScript1.py

2013-09-25 08:12:26,087 DEBUG [AIF]: CUSTOM_LOCATION = 1

2013-09-25 08:12:26,087 DEBUG [AIF]: CUSTOM_YEAR = 2013

2013-09-25 08:12:26,088 DEBUG [AIF]: Custom script parameter values by name: begin

2013-09-25 08:12:26,091 DEBUG [AIF]: The value of parameter CUSTOM_LOCATION is = 1

2013-09-25 08:12:26,093 DEBUG [AIF]: The value of parameter CUSTOM_YEAR is = 2013

2013-09-25 08:12:26,094 DEBUG [AIF]: Custom script parameter values by name: end

2013-09-25 08:12:26,535 DEBUG [AIF]: Submitting a data rule via a script using the
following command: C:/Oracle/Middleware/user_projects/epmsystem1/
FinancialDataQuality/loaddata.bat "admin" "*****" "SRESBA1_DR1" "Y" "N" "STORE_DATA"
"SNAPSHOT" "N" "Jan-2003" "Jan-2003" "ASYNC"

2013-09-25 08:12:26,596 DEBUG [AIF]: Data rule submitted fine.

2013-09-25 08:12:26,635 INFO [AIF]:
=====

2013-09-25 08:12:26,636 INFO [AIF]: Custom Script: end

2013-09-25 08:12:26,637 INFO [AIF]:
=====

```

Submitting a Report

This example shows how to submit a report. The script calls a batch file that includes the necessary parameters instead of including the parameters directly in the script. Note that if you try to submit a report script with an equal sign (=) in the parameters, the script fails.

```
#
```

This sample Jython script provides code related to custom scripts. All the messages being logged (meaning printed) are visible in the process log file as per Log Level profile.

```
#

# Log the beginning of the script, at the INFO level

fdmAPI.logInfo("=====
=)

fdmAPI.logInfo("Custom Script: Begin")

fdmAPI.logInfo("=====
=)

# Execute the subprocess to call and run the batch file

import os

import subprocess

os.chdir("D:/ORCL/MW/EPMSys11R1/products/FinancialDataQuality/bin")

myScriptName = "D:/ORCL/MW/EPMSys11R1/products/FinancialDataQuality/bin/
myreport.bat"

retcode = subprocess.Popen(myScriptName)

fdmAPI.logDebug("The return code = " + retcode)

The myreport.bat file referenced above can contain the equal sign as a parameter as
in the following:

runreport.bat <username> <password> "TB All Columns (Per,Cat,Loc)" "Period=Jan 15"
"Category=Actual" "Location=AAA" "Rule Name=AAA" "Report Output Format=PDF"
```

Using the JAVA API

This section explains how to use JAVA API with Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

Overview

Oracle Hyperion Financial Data Quality Management, Enterprise Edition automatically initializes the data load workflow context information prior to invoking the Import, Mapping and Event scripts. The `fdmContext` object is initialized with the list of properties listed below. The properties can be accessed by referencing `fdmContext["<PROPERTY NAME>"]`. For example, to access Location Name, use `fdmContext["LOCNAME"]`. To execute a script for a specific data load, write something like the following:

```
if fdmContext["LOCNAME"] == "ITALY":

    Write script logic here
```

FDM object properties include:

- APPNAM
- APPID
- BATCHSCRIPTDIR
- CATKEY
- CATNAME
- CHKSTATUS
- EPMORACLEHOME
- EPMORACLEINSTANCEHOME
- EXPORTFLAG
- EXPORTMODE
- EXPSTATUS
- FILEDIR
- FILENAME
- IMPORTFLAG
- IMPORTFORMAT
- IMPORTMODE
- IMPST
- IMPSTATUS
- INBOXDIR
- LOADID
- LOCKEY
- LOCNAME
- MULTIPERIODLOAD
- OUTBOXDIR
- PERIODNAME
- PERIODKEY
- PROCESSSTATUS
- RULEID
- RULENAME
- SCRIPTSDIR
- SOURCENAME
- SOURCETYPE
- TARGETAPPDB
- TARGETAPPNAME
- VALSTATUS

JAVA API List

Table 7-3 JAVA API List

API	Description
BigDecimal getPOVLocation(BigDecimal pLoadId)	Returns the Partition Key based on the LOADID.
BigDecimal getPOVCategory(BigDecimal pLoadId)	Returns the Category Key based on the LOADID.
Date getPOVStartPeriod(BigDecimal pLoadId)	Returns the Period Key of the start period based on the LOADID.
Date getPOVEndPeriod(BigDecimal pLoadId)	Returns the Period Key of the end period based on the LOADID when loading a single period, and the start period and end period are the same.
executePLSQL	Executes a block of Procedural Language/Structured Query Language (PL/SQL) code. This API takes the following three parameters: - String—A valid block of PL/SQL code. The PL/SQL block must be enclosed within double quotes. - Object array—An input array of Java objects (e.g. String, BigDecimal) representing input parameters to the block of code. The object array is used to insert into the PL/SQL where there are place holders (question marks). - Boolean—whether to perform a commit or not The Boolean value is either True or False. An example of how to execute the PL/SQL: <pre>fdmAPI.executePLSQL("BEGIN dbms_stats.gather_table_stats(user,?,estimate_percent=>dbms_stats.auto_sample_size,force=>TRUE); END;","["tDataSeg_T"]",True);</pre>

Table 7-3 (Cont.) JAVA API List

API	Description
getBatchJobDetails	The getBatchJobDetails returns the following column for each job submitted by the batch: - BATCH_ID - BATCH_NAME - APPLICATION_ID - BATCH_TYPE - BATCH_EXECUTION_MODE - BATCH_WAIT_TIMEOUT - USER_POV_PERIOD - OPEN_BATCH_FOLDER - PLAN_TYPE - FILENAME_SEPARATOR - BATCH_GROUP_ID - BEF_PROCESS_BATCH_SCRIPT - AFT_PROCESS_BATCH_SCRIPT - EXECUTION_DATE - EXECUTED_BY - LOADID - BATCH_LOADID - PARENT_BATCH_LOADID - PARTITIONKEY - CATKEY - START_PERIODKEY - END_PERIODKEY - IMPORT_FROM_SOURCE_FLAG - EXPORT_TO_TARGET_FLAG - RECALCULATE_FLAG - CHECK_FLAG - JOURNAL_FLAG - IMPORT_MODE - EXPORT_MODE - IMPGROUPKEY - PROCESS_NAME - RULE_TYPE - LOG_FILE - OUTPUT_FILE - EPM_ORACLE_INSTANCE - ODI_SESSION_NUMBER - STATUS
int executedDML(String query,Object[] parameters)	Execute any DML Statement. Provide the query and parameter. The parameter is provided as a list.
logDB(String pEntityType, String pEntityName, int pLogSequence, String pLogMessage)	Log messages to a database table AIF_PROCESS_LOGS.
logFatal(String pLogMessage)	Log a message when a fatal error occurs. This message is displayed at all log levels.
logError(String pLogMessage)	Log a message when an error occurs. This message is displayed at log level 2 or higher.

Table 7-3 (Cont.) JAVA API List

API	Description
<code>logWarn(String pLogMessage)</code>	Log a message when a warning condition error occurs. This message is displayed at log level 3 or higher.
<code>logInfo(String pLogMessage)</code>	Log an informational message. This message is displayed at log level 4 or higher.
<code>logDebug(String pLogMessage)</code>	Log a debug message. This message is displayed at log level 5.
<code>Map getProcessStates(BigDecimal pLoadId)</code>	Returns status of Workflow process. Available statuses: • IMPSTATUS—Returns the status of the import process. A 0 status indicates that the process has not yet started, or the process failed. A 1 status indicates the process is successful. • VALSTATUS—Returns the status of validation process. A 0 status indicates that the process has not yet started, or the process failed. A 1 status indicates the process is successful. • EXPSTATUS—Returns the status of export process. A 0 status indicates that the process has not yet started, or the process failed. A 1 status indicates the process is successful. • CHKSTATUS—Returns the status of check process. A 0 status indicates that the process has not yet started, or the process failed. A 1 status indicates the process is successful. • PROCESSSTATUS—Returns the exact error code. The detail for the error code can be found in the <code>tlogprocessstates</code> table.
<code>Map getPeriodDetail(Date pPeriodKey,String pApplicationName) //returns PERIODTARGET and YEARTARGET</code>	Returns the Target period mapping for a given Period key.
<code>Object getCustomScriptParameterValue(BigDecimal pLoadId,String pParameterName)</code>	Returns the value for given custom script parameter name and loadID.
<code>Object getCustomScriptParameterValue(String pParameterName)</code>	Returns the value for given custom script parameter name and context initialized loaded.
<code>ResultSet getBatchDetails()</code>	Returns batch definition information from AIF_BATCHES table. The fields returned by <code>getBatchDetails()</code>: • BATCH_ID • BATCH_NAME • APPLICATION_ID • BATCH_TYPE • BATCH_EXECUTION_MOD

Table 7-3 (Cont.) JAVA API List

API	Description
ResultSet getBatchJobDetails(BigDecimal pLoadId)	Retrieves error messages logged to the database table AIF_PROCESS_LOGS for the given loadid.
ResultSet getCategoryList()	Returns a list of Categories in a result set. The fields returned by getCategoryList(): • CATKEY • CATNAME
ResultSet getCheckEntityGroupList(BigDecimal pApplicationId)	Returns a list of Check Groups in a result set.
ResultSet getCheckEntityForGroup(String pValGroupKey)	Return a list of entities in a Check Group in a result set.
ResultSet getCheckEntityGroupList(BigDecimal pApplicationId)	Return a list of Check Rule Groups in a result set.
ResultSet getCheckEntityForGroup	Return a list of Check Rule Group rules in a result set.
ResultSet getCustomDBLog()	Retrieve error messages logged to the database table AIF_PROCESS_LOGS for the current process. The fields returned by getCustomDBLog(): • ENTITY_TYPE • ENTITY_NAME • LOG_SEQUENCE
ResultSet getCustomDBLog(BigDecimal pLoadId)	Returns the log statements from DB for a given loadID.
ResultSet getCustomScriptParameters()	Returns the list of custom script parameters in a result set for the context initialized loadID.
ResultSet getCustomScriptParameters(BigDecimal pLoadId)	Returns the list of custom script parameters in a resultset for the given loadID.
ResultSet getPeriodList()	Returns a list of Periods in a result set. The fields returned by getPeriodList(): • PERIODKEY • PERIODDESC
ResultSet executeQuery(String query, Object[] parameters)	Execute any SQL Query Statement. The query results are returned in a result set. Provide the query and parameter. The parameter is provided as a list.

Table 7-3 (Cont.) JAVA API List

API	Description
ResultSet getImportFormatDetails(String pImpGroupKey)	Return the Import Format details in a result set based on the Import Format key. The fields returned by getImportFormatDetails(String pImpGroupKey): • IMPGROUPKEY • IMPGROUPDESC • IMPGROUPFILETYPE • IMPGROUPDELIMITER • IMPGROUPTYPE • IMPSOURCESYSTEMID • IMPSOURCELEDGERID • IMPSOURCECOAID • IMPTARGETAPPLICATIONID • IMPADAPTERID • IMPDRILLURLID • IMPODISCENARIO • IMPREGENSCEN • IMPDRILLREQUESTMETHOD • IMPDRILLURL • IMPTARGETSOURCESYSTEMID
ResultSet getImportFormatMapDetails(String pImpGroupKey)	Return the Import Format Mapping details in a result set for a given Import Format key. This currently supports only file based import formats. The fields returned by getImportFormatMapDetails(String pImpGroupKey): • IMPSEQ • IMPGROUPKEY • IMPFLDFIELDNAME • IMPFLDFIXEDTEXT • IMPFLDSTARTPOS • IMPFLDLENGTH • IMPFLDSOURCECOLNAME

Table 7-3 (Cont.) JAVA API List

API	Description
ResultSet getLocationDetails(BigDecimal pPartitionKey)	Return the Location details in a record set for a given Location key. The fields returned by getLocationDetails: • PARTITIONKEY • PARTNAME • PARTDESC • PARTNOTES • PARTLASTIMPFIL • PARTLASTEXPFILE • PARTIMPGROUP • PARTLOGICGROUP • PARTVALGROUP • PARTVALENTGROUP • PARTCURRENCYKEY • PARTPARENT • PARTTYPE • PARTSEQMAP • PARTDATAVALUE • PARTSEGMENTKEY • PARTCONTROLSTYPE • PARTCONTROLSGROUP1 • PARTCONTROLSGROUP2 • PARTCONTROLSAPPROVER • PARTCONTROLSAPPROVERPROXY • PARTCONTROLSREDFLAGLEVEL • PARTCLOGICGROUP • PARTINTGCONFIG1 • PARTINTGCONFIG2 • PARTINTGCONFIG3 • PARTINTGCONFIG4 • PARTADAPTOR • PARTSOURCESYSTEMID • PARTSOURCELEDGERID • PARTTARGETAPPLICATIONID • PARTPARENTKEY • PARTSOURCEAPPLICATIONID • PARTTARGETSOURCESYSTEMID For example if you want to know the fields returned by the getLocationDetails API, run the following script: <pre>rs = fdmAPI.getLocationDetails(fdmContext["LOC KEY"]) i = 1 metaData = rs.getMetaData() while i <= metaData.getColumnCount():</pre>

Table 7-3 (Cont.) JAVA API List

API	Description
	<pre>fdmAPI.logDebug(metaData.getColumnLabel(i))</pre>
	<pre>i +=1</pre>

Table 7-3 (Cont.) JAVA API List

API	Description
ResultSet getRuleDetails(BigDecimal pRuleId)	Returns the Data Rule details in a record set for a given Data Rule ID. The fields returned by getRuleDetails(BigDecimal pRuleId): • RULE_ID • SOURCE_SYSTEM_ID • SOURCE_LEDGER_ID • APPLICATION_ID • RULE_NAME • RULE_DESCRIPTION • PLAN_TYPE • LEDGER_GROUP • INCL_ZERO_BALANCE_FLAG • BALANCE_SELECTION • AMOUNT_TYPE • BALANCE_METHOD_CODE • BALANCE_TYPE • BAL_SEG_VALUE_OPTION_CODE • EXCHANGE_RATE_OPTION_CODE • EXCHANGE_BEGIN_RATE_TYPE • EXCHANGE_END_RATE_TYPE • EXCHANGE_AVERAGE_RATE_TYPE • DATA_SYNC_OBJECT • DATA_SYNC_OBJECT_ID • PARTCONTROLSAPPROVERPROXY • PARTCONTROLSREDFLAGLEVEL • STATUS • PARTITIONKEY • CATKEY • INCLUDE_ADJ_PERIODS_FLAG • BALANCE_AMOUNT_BS • BALANCE_AMOUNT_IS • AS_OF_DATE • BLANK_PERIODKEY • BR_MEMBER_NAME • BR_MEMBER_DISP_NAME • CALENDAR_ID • CURRENCY_CODE • DP_MEMBER_NAME • FILE_NAME_DATE_FORMAT • FILE_NAME_STATIC • FILE_NAME_SUFFIX_TYPE • FILE_PATH • LEDGER_GROUP_ID • PERIOD_MAPPING_TYPE • VERSION • SIGNAGE_METHOD • DIRECT_FILE_LOAD_FLAG • LOAD_OPTIONS

Table 7-3 (Cont.) JAVA API List

API	Description
	• RULE_ATTR1 • RULE_ATTR2 • RULE_ATTR3 • RULE_ATTR4 • ICP_LOAD • MULTI_PERIOD_FILE_FLAG • IMPGROUPKEY • SOURCE_APP_PLAN_TYPE
showCustomDBLog()	Show a list of custom messages in the user interface after completion of a process. Message can be displayed at the end of a data load workflow step like import, validate, export, check or at the end of a custom script execution. Note messages are displayed only when the process are run in online mode.
showCustomFile(String filePath)	Show a custom file (log file, report file) in the user interface after completion of a process. Message can be displayed at the end of a data load workflow step like import, validate, export, check or at the end of a custom script execution. Note messages are displayed only when the process are run in online mode.
showCustomMessage(String message)	Show a custom message in the user interface after completion of a process. Message can be displayed at the end of a data load workflow step like import, validate, export, check or at the end of a custom script execution. Note messages are displayed only when the process are run in online mode.
String getCategoryMap(BigDecimal pCatKey,String pApplicationName)	Returns the Scenario for a given Category and Application Name.
String getCustomMessage()	Retrieves the last custom message raised for the current process.
String getCustomMessage(BigDecimal pLoadId)	Retrieves the last custom message raised for the given loadid.
String getCustomFile()	Retrieves the custom file created for the current process.
String getCustomFile(BigDecimal pLoadId)	Retrieves the custom file created for the given loadid.
String getPOVDataValue(BigDecimal pPartitionKey)	Returns the data value of the Location.
String getDirTopLevel(BigDecimal pApplicationId)	Returns the Top Level directory based on the Application.
String getDirInbox(BigDecimal pApplicationId)	Returns the Inbox directory based on the Application.
String getDirOutbox(BigDecimal pApplicationId)	Returns the Outbox directory based on the Application.
String getDirScripts(BigDecimal pApplicationId)	Returns the Scripts directory based on the Application.

Table 7-3 (Cont.) JAVA API List

API	Description
String getProcessLogFile(BigDecimal pLoadId)	Returns the log file name for the given loadID.
String getProfileOptionValue(String pProfileOptionName, BigDecimal pApplicationId, String pUserName)	Returns the value set for an option. Options can be set at the System Setting, Application Setting, and User Setting. The order of precedence is: User, Application, and System. The API determines the appropriate applicable value and returns the value.
void writeToProcessLogsDB(BigDecimal pLoadId, String pEntityType, String pEntityName, int pLogSequence, String pLogMessage)	Writes the log information to the AIF_PROCESS_LOGS table. Use Entity Type and Entity Name to group the logged messages. Logs can be created only from a data load workflow process.
void writeToProcessLogsFile(BigDecimal pLoadId, String pLogMessage)	Writes the log information to the Data Load Process log file. The logs is written based on the process log level. Logs can be created only from a data load workflow process.
 Note: It is recommended that you use the logging API, for example logDebug or, logInfo API, instead of using the writeToProcessLogsFile API.	
void closeConnection()	Use to close the database connection.
void closeResultSet(ResultSet resultSet)	Use to close result object.
void updateImportFormat(String pImpgroupKey, BigDecimal pLoadId)	Update the import format for the current run. This is applicable only for File-based import formats.

Working with UNICODE Characters in Jython Scripts

When writing script code in Jython technology, specify any non-English strings in UNICODE by prefixing the letter "u" before the string in double quotes. This means instead of defining a string as "MyValue" define it as u"MyValue." See the following example used in data load mapping script for the Account dimension:


```

entity = fdmRow.getString("ENTITY")
account = fdmRow.getString("ACCOUNT")
if (entity == u"グローバル" and account == "1110"):
    fdmResult = u"Global テスト"
elif (entity == u"ローカル" and account == "1110"):
    fdmResult = u"Local テスト"
else:
    fdmResult = account

```

The scripts above uses the "u" prefix for the user defined strings. You can optionally specify the u prefix for English/ASCII strings (that is, you can use "1110" or u"1110"). The following shows the result of the mapping applied on the workbench.

Load Data				
View ▾	Format ▾	Show ▾	Download Template ▾	
			1110*	
	Source-Company	Entity	Source-Account	Account
--	ローカル	SRE_ローカル	1110	Local テスト
--	グローバル	SRE_グローバル	1110	Global テスト
--	ローカル	SRE_ローカル	1110	Local テスト

Using JAVA IDE to Develop Scripts

You can use popular Java IDE tools like Oracle jDeveloper, or Eclipse to develop and test scripts. Before using Eclipse to develop scripts you must install and configure the PyDev Interpreter. Refer to <http://pydev.org> for more details. After you have configured the IDE environment, copy the following JAR files from the EPM Server where Oracle Hyperion Financial Data Quality Management, Enterprise Edition is installed (File location EPM_ORACLE_HOME/products/FinancialDataQuality/lib):

1. aif-apis.jar
2. aif-custom.jar

In addition, download appropriate (Oracle or SQL Server) JDBC driver JAR. After you have copied these files to the Project working directory, include them in the Project you create. Below is sample of the initialization steps required when running from your selected IDE:

```

#Start Initialize Code

#Required for Dev Mode. Not required in production script

import java.math.BigDecimal as BigDecimal

import java.sql as sql

import com.hyperion.aif.scripting.API as API

fdmAPI = API()

conn = None

```

```
conn = sql.DriverManager.getConnection("jdbc:oracle:thin:@server:1521:orcl", "user",  
"password");  
  
conn.setAutoCommit(False)  
  
fdmAPI.initializeDevMode(conn);  
  
print "SUCCESS CONNECTING TO DB"  
  
fdmContext = fdmAPI.initContext(BigDecimal(1720))  
  
#End Initialize Code Required for Dev Mode. Not required in production script  
  
#Code to be added to production script  
  
print fdmContext["LOCNAME"]  
  
print fdmContext["LOCKEY"]  
  
print fdmContext["APPID"]
```

Using Visual Basic API and Samples

This section explains how to use the Visual Basic API with Oracle Hyperion Financial Data Quality Management, Enterprise Edition. It includes samples.

Overview

This section includes:

- [Registering the Visual Basic API DLL Manually](#)
- [Visual Basic API List](#)
- [Visual Basic Sample Scripts](#)

Registering the Visual Basic API DLL Manually

If you use the Visual Basic API, then you must register the Visual Basic API DLL manually. Oracle Hyperion Financial Data Quality Management, Enterprise Edition does not register the Visual Basic API DLL at installation.

To register the Visual Basic API DLL manually:

1. Open a command window.
2. Navigate to: EPM_ORACLE_HOME\products\FinancialDataQuality\lib\Windows
3. Execute **RegisterVBAPI.vbs**.

The RegisterVBAPI.vbs script requires one parameter to be passed to it. The parameter must be the path (wrapped in quotes) to the Oracle Instance Path.

For example, specify RegisterVBAPI.vbs "C:\Oracle\Middleware\user_projects\epmsystem1"

Visual Basic API List

The API list below shows the objects you can use to programmatically extend Oracle Hyperion Financial Data Quality Management, Enterprise Edition using Visual Basic.

API Class (clsAppServerDM)

fdmAPI.API Call

Table 7-4 API Class (clsAppServerDM)

API Call	Method/Property/Object Reference	Description
fnInitializeMainAPI	Method	Initialize the VB API. This function must be called first in order to use the VB API. Input: ByVal LoadID As String, ByVal Token As String, ByVal OracleInstancePath As String, ByVal JavaHomePath As String, ByVal DatabaseProvider As String Output: True or False
URLDecodeString	Method	Decode a URL Encoded string. Input: ByVal value as String Output: String
Dispose	Method	Initiates the .Net garbage collection process to destroy the API object.

Application Class (clsApp)

fdmAPI.API.DataWindow.Connection.PobjApp.API Call

Table 7-5 Application Class (clsApp)

API Call	Method/Property/Object Reference	Description
PstrConnInfoCacheFile	Property	Returns the path to the ccs file stored in the user projects directory under the FDM folder. This file contains the connection information returned by COMJNI. This file is encrypted.
PstrConnInfoTempFilePath	Property	Path of ccs file only
PstrDatabaseName	Property	Database name
PstrDatabasePassword	Property	Database password
PstrDatabaseServer	Property	Database server
PstrDatabaseUserName	Property	Database user name
PstrDataDir	Property	Data directory for the LoadID
PstrDBConnectionString	Property	DB connection string returned from COMJNI

Table 7-5 (Cont.) Application Class (clsApp)

API Call	Method/Property/Object Reference	Description
PstrDBPort	Property	DB port
PstrDBProvider	Property	DB provider
PstrInboxDir	Property	Inbox directory for Load ID
PstrJavaHomePath	Property	Java Home Path Load ID
PstrLoadID	Property	Load ID
PstrOracleHomePath	Property	Oracle Home
PstrOracleInstancePath	Property	Oracle Instance Path
PstrOutboxDir	Property	Outbox directory for Load ID
PstrScriptsDir	Property	Scripts directory for Load ID
PstrToken	Property	Token
PstrTopLevelDir	Property	Top level directory for Load ID

Connection Class (clsConnection)

fdmAPI.API.DataWindow.Connection.API Call

Table 7-6 API.DataWindow.Connection

API Call	Method/Property/Object Reference	Description
GlngErrNo	Property	Contains the error number in the event of an error.
GstrError	Property	Contains the error number in the event of an error.
PblnEncodingUnicode	Property	Returns "True" if set to UTF-8, but "false" for all others.
PblnUseTrans	Property	This property defaults to "False." If it is set to "True," the query executed in ExecuteDML is executed as a transaction.
PobjApp	Object Reference	Reference to the clsAPP class
PstrConnection	Property	Returns the connection string to Oracle Hyperion Financial Data Quality Management, Enterprise Edition.
PstrDatabase	Property	Connection string database catalog name
PstrDatabaseUserID	Property	Contains the database user ID.
PstrDirData	Property	Returns the path to the Data directory.
PstrDirInbox	Property	Returns the path to the Inbox directory.
PstrDirOutbox	Property	Returns the Outbox directory based on LoadID.

Table 7-6 (Cont.) API.DataWindow.Connection

API Call	Method/Property/Object Reference	Description
PstrDirReport	Property	Returns blank. Needs to point to where the "templates" directory is installed.
PstrDirScripts	Property	Returns Scripts directory for LoadID.
PstrLanguageCode	Property	Returns FDMEE API language code. Returns "en" (for English) only. This property cannot be changed in the user interface.
PstrProvider	Property	Returns the provider used by FDMEE API.
PstrQueryDateMask	Property	Get/Set Query Date mask which is used through the API for queries involving date fields. The default is "yyyMMdd."
PstrResourceFileDirectory	Property	Points to the FDM API localization files directory path.
PstrServer	Property	Prints the database server name.
fGenerateRnd	Method	Returns the random number (short type).
fGetOSUserID	Method	Returns the OS user of the current process.

DataManipulation Class

fdmAPI.API.DataWindow.DataManipulation.API Call

Table 7-7 DataManipulation Class

API Call	Method/Property/Object Reference	Description
fExecuteDML	Method	Execute DML Query. Inputs: ByVal strSQL As String Output: Boolean
PIngRecordsAffected	Property	Returns the number of records affected by the query executed in fExecuteDML.

Utilities Class (clsUtilities.cls)

fdmAPI.API.DataWindow.Utilities.API Call

Table 7-8 Utilities Class (clsUtilities.cls)

API Call	Method/Property/Object Reference	Description
fParseString	Method	Extract a substring from a delimited string based on its field position within the total number of fields. Inputs: ByVal strRecord As String, ByVal intFldCount As Integer, ByVal intFldToReturn As Integer, ByVal strDelimiter As String Output: String
fGetRandomNumber	Method	Get a random number based on a two integer range. Inputs: ByRef intUpper As Short, ByRef intLower As Short Output: Short

Data Access Class (clsDataAccess)

fdmAPI.API.DataWindow.DataAccess.API Call

Table 7-9 Data Access Class (clsDataAccess)

API Call	Method/Property/Object Reference	Description
farsFireHose	Method	Open an ADO forward only/read only recordset for fast data streaming. Inputs: ByVal strSQL As String, Optional ByVal blnServerCursor As Boolean, Optional ByVal lngMarshalType As Long • strSQL—SQL statement to execute • blnServerCursor—Optional, Cursor location switch (True=Database Server,False=Client) • lngMarshalType—Optional, Value used to control how the recordset is returned Output: ADODB Recordset

Table 7-9 (Cont.) Data Access Class (clsDataAccess)

API Call	Method/Property/Object Reference	Description
farsKeySet	Method	Open an ADO keyset type recordset based on the specified lock type. Inputs: Inputs: ByVal strSQL As String, Optional ByVal lngLockType As Long, Optional ByVal blnServerCursor As Boolean, Optional ByVal lngMarshalType As Long • strSQL—SQL statement to execute • lngLockType—ADO LockTypeEnum (adLockReadOnly=1;adLockPessimistic=2;adLockOptimistic=3;adLockBatchOptimistic=4) • blnServerCursor—Optional, cursor location switch (True=Database Server,False=Client) • lngMarshalType—Optional, Value used to control how the recordset is returned. Output: ADODB.Recordset
farsPage	Method	Open an ADO BatchOptimistic recordset limited to the records found on the specified page. The record set returned is programmatically created and is disconnected. Inputs: ByVal strSQL As String, ByVal lngPageSize As Long, ByVal lngPage As Long, lngOPRecordCount As Long, Optional ByVal lngMarshalType As Long • strSQL—SQL statement to execute • lngPageSize—Number of records per page • lngPage—Desired page number • lngOPRecordCount—Output parameter containing record set record count • lngMarshalType—Optional, Value used to control how the record set is returned Output: ADODB.Recordset

Table 7-9 (Cont.) Data Access Class (clsDataAccess)

API Call	Method/Property/Object Reference	Description
farsPageFind	Method	Open an ADO BatchOptimistic recordset limited to the records found on the specified page and for the specified criteria. The recordset returned is programmatically created and is disconnected. Inputs: ByVal strSQL As String, ByVal strCriteria As String, ByVal lngStartPosition As Long, ByVal lngPageSize As Long, lngOPRecordCount As Long, lngOPFindPostion As Long, Optional ByVal lngMarshalType As Long • strSQL—SQL statement to execute • strCriteria—ADO recordset FIND criteria statement (See ADO FIND method) • lngStartPosition—Record index to begin the search • lngPageSize—Number of records per page • lngPage—Desired page number • lngOPRecordCount—Output parameter containing recordset record count • lngOPFindPostion—Output parameter containing position where record was found • lngMarshalType—Optional, Value used to control how the recordset is returned Output: ADODB.Recordset

Table 7-9 (Cont.) Data Access Class (clsDataAccess)

API Call	Method/Property/Object Reference	Description
farsPageFindFromKey	Method	Open an ADO BatchOptimistic recordset limited to the records found on the specified page and for the specified criteria. The recordset returned is programmatically created and is disconnected. Inputs: ByVal strSelect As String, ByVal strFrom As String, ByVal strWhere As String, ByVal strGroupBy As String, ByVal strHaving As String, ByVal strOrderBy As String, ByVal strKeyFieldName As String, ByVal strFindFieldName As String, ByVal strCriteria As String, ByVal lngStartPosition As Long, ByVal lngPageSize As Long, lngOPRecordCount As Long, lngOPFindPostion As Long, Optional ByVal • lngMarshalType—As Long • strSelect—Select clause of SQL statement to execute • strFrom—From clause of SQL statement to execute • strWhere—Where clause of SQL statement to execute • strGroupBy GroupBy—Clause of SQL statement to execute • strHaving—Having clause of SQL statement to execute • strOrderBy OrderBy—Clause of SQL statement to execute • strKeyFieldName—Name of key field used for fast page location • strFindFieldName—Name of the field used in the find criteria • strCriteria—Find criteria statement • lngStartPosition—Record index to begin the search • lngPageSize—Number of records per page • lngPage—Desired page number • lngOPRecordCount—Output parameter containing recordset record count • lngOPFindPostion—Output parameter containing position where record was found

Table 7-9 (Cont.) Data Access Class (clsDataAccess)

API Call	Method/Property/Object Reference	Description
farsPageFromKey	Method	• lngMarshalType—Optional, Value used to control how the recordset is returned Output: ADODB.Recordset Open an ADO BatchOptimistic recordset limited to the records found on the specified page. This method is optimized for high volume queries. The recordset returned is programmatically created and is disconnected. Inputs: ByVal strSelect As String, ByVal strFrom As String, ByVal strWhere As String, ByVal strGroupBy As String, ByVal strHaving As String, ByVal strOrderBy As String, ByVal strKeyFieldName As String, ByVal lngPageSize As Long, ByVal lngPage As Long, lngOPRecordCount As Long, Optional ByVal lngMarshalType As Long • strSelect—Select clause of SQL statement to execute • strFrom—From clause of SQL statement to execute • strWhere—Where clause of SQL statement to execute • strGroupBy—GroupBy clause of SQL statement to execute • strHaving—Having clause of SQL statement to execute • strOrderBy—OrderBy clause of SQL statement to execute • strKeyFieldName—Name of key field used for fast page location • lngPageSize—Number of records per page • lngPage—Desired page number • lngOPRecordCount—Output parameter containing recordset record count • lngMarshalType—Optional, Value used to control how the recordset is returned Output: ADODB.Recordset

Table 7-9 (Cont.) Data Access Class (clsDataAccess)

API Call	Method/Property/Object Reference	Description
farsSnap	Method	Open an ADO static type read only recordset. Inputs: ByVal strSQL As String, Optional ByVal blnServerCursor As Boolean, Optional ByVal lngMarshalType As Long • strSQL—SQL statement to execute • blnServerCursor—Optional, cursor location switch (True=Database Server,False=Client) • lngMarshalType—Optional, Value used to control how the recordset is returned Outputs:ADODB.Recordset
farsTableAppend	Method	Open an ADO keyset type recordset based on the specified lock type using the ADO "Table Direct "command directive and with the "Append-Only Rowset " property set to true. Inputs: ByVal strTableName As String, Optional ByVal lngMarshalType As Long • strTableName—Name of table to open • lngMarshalType—Optional, Value used to control how the recordset is returned Outputs: ADODB.Recordset

SQL Manager Class (clsSQLMgr)

fdmAPI.API.SQLMgr.API Call

Table 7-10 SQL Manager Class (clsSQLMgr)

API Call	Method/Property/Object Reference	Description
fImpFieldsForGroup	Method	Returns a recordset containing all of the records ("Mappings") for the Import Format name (string) passed in at runtime. Inputs: ByVal strGroupKey As String, Optional ByVal blnReadOnly As Boolean = False, Optional ByVal blnSQLOnly As Boolean = False, Optional ByVal lngMarshalType As Integer = 0 • strGroupKey—Name of Import group key to return field parsing instructions • blnReadOnly—Optional, Flag to create a read only recordset (Type=farsSnap) • blnSQLOnly—Optional, Switch for generating SQL only • lngMarshalType—Optional, Value used to control how the recordset is returned.
fImpGroup	Method	Returns a recordset containing all records in TBHVIMPGROUP table for the Import Format name (string) passed in. Inputs: ByVal strGroupKey As String, Optional ByVal blnReadOnly As Boolean = False, Optional ByVal blnSQLOnly As Boolean = False, Optional ByVal lngMarshalType As Integer = 0 • strGroupKey—Name of Import group key to return field parsing instructions for • blnReadOnly—Optional, Flag to create a read only recordset (Type=farsSnap) • blnSQLOnly—Optional, Switch for generating SQL only • lngMarshalType—Optional, Value used to control how the recordset is returned

Table 7-10 (Cont.) SQL Manager Class (clsSQLMgr)

API Call	Method/Property/Object Reference	Description
fImpGroupList	Method	Returns a recordset containing all import format groups based on group type. Inputs: ByVal lngGroupType As Integer, Optional ByVal blnIncludeNone As Boolean = False, Optional ByVal blnReadOnly As Boolean = False, Optional ByVal blnSQLOnly As Boolean = False, Optional ByVal lngMarshalType As Integer = 0 Output: ADODB.Recordset
fIntDimensionList	Method	Returns a recordset of dimensions used in the Target Application. ByVal strApplicationID As String, Optional ByVal blnReadOnly As Boolean = False, Optional ByVal blnSQLOnly As Boolean = False, Optional ByVal lngMarshalType As Integer = 0 Output: ADODB.Recordset
fLoadDetails	Method	Return load details for specified Load ID. Inputs: ByVal strLoadID As String, Optional ByVal blnReadOnly as Boolean = False, Optional ByVal blnSQLOnly as Boolean = False, Optional ByVal lngMarshalType as Integer = 0 Output: ADODB.Recordset
fPartition	Method	Returns a recordset containing all of the properties of the location passed into the function. ByVal strLoc As String, Optional ByVal blnReadOnly As Boolean = False, Optional ByVal blnSQLOnly As Boolean = False, Optional ByVal lngMarshalType As Integer = 0 Output: ADODB.Recordset
fPartitionList	Method	Returns a recordset containing all properties of all locations that exist. Inputs: Optional ByVal strLocNoRelatives As String = "", Optional ByVal strLocChildren As String = "", Optional ByVal blnNoParent As Boolean = False, Optional ByVal blnReadOnly As Boolean = False, Optional ByVal blnSQLOnly As Boolean = False, Optional ByVal lngMarshalType As Integer = 0 Output: ADODB.Recordset

Table 7-10 (Cont.) SQL Manager Class (clsSQLMgr)

API Call	Method/Property/Object Reference	Description
fPeriodKeyList	Method	Returns a recordset containing the list of period keys for the loadid passed to the function. Inputs: ByVal strLoadID As String, Optional ByVal blnReadOnly As Boolean = False, Optional ByVal blnSQLOnly As Boolean = False, Optional ByVal lngMarshalType As Integer = 0 Output: ADODB.RecordsetOutput: ADODB.Recordset
fPOVDataValue	Method	Return the data value for the specified partition key. Inputs: ByVal strPartitionKey As String, Optional ByVal blnReadOnly As Boolean = False, Optional ByVal blnSQLOnly As Boolean = False, Optional ByVal lngMarshalType As Integer = 0 Output: ADODB.Recordset
fProcessBallImpFile	Method	Return a recordset containing the name of the imported file for a given load id. For all non-file based imports, this function returns an empty string. Inputs: ByVal strLoadID As String, Optional ByVal blnReadOnly As Boolean = False, Optional ByVal blnSQLOnly As Boolean = False, Optional ByVal lngMarshalType As Integer = 0 Output: ADODB.Recordset
fProcessLogForPOV	Method	Return a single process status record for the specified POV value/ intersection. Inputs: ByVal strLoc As String, ByVal strCat As String, ByVal strPer As String, ByVal strRuleID As String, Optional ByVal blnReadOnly As Boolean = False, Optional ByVal blnSQLOnly As Boolean = False, Optional ByVal lngMarshalType As Integer = 0 Output: ADODB.Recordset

Table 7-10 (Cont.) SQL Manager Class (clsSQLMgr)

API Call	Method/Property/Object Reference	Description
fTrialBalanceConverted	Method	Returns a recordset containing all records pulled from: tDataSeg based on LoadID, Loc, Scenario, and Period. Inputs: ByVal strLoadID As String, ByVal strLoc As String, ByVal strCat As String, ByVal strPer As String, Optional ByVal blnExcludeIgnore As Boolean = False, Optional ByVal blnSQLOnly As Boolean = False, Optional ByVal lngPageSize As Integer = 0, Optional ByVal lngPage As Integer = 0, Optional ByRef lngOPRecordCount As Object = Nothing, Optional ByVal lngMarshalType As Integer = 0 Output: ADODB.Recordset
fTrialBalanceLite	Method	Return a recordset listing the DataKey field from the data segment table assigned to the specified location. The data is returned based on the specified POV values and calculation types. The purpose of this recordset is to create a fast/light weight recordset that you can use to execute a looping cursor delete. Input: ByVal strLoadID As String, ByVal strLoc As String, ByVal strCat As String, ByVal strPer As String, ByVal strCalcTypes As String, Optional ByVal lngStartCalcType As Integer = 0, Optional ByVal lngEndCalcType As Integer = 0, Optional ByVal strJournalID As String = "", Optional ByVal blnSQLOnly As Boolean = False, Optional ByVal lngMarshalType As Integer = 0 Output: ADODB.Recordset
fValEntGroupList	Method	Returns a recordset containing all validation entity group records. Input: Optional ByVal blnIncludeNone As Boolean = False, Optional ByVal blnReadOnly As Boolean = False, Optional ByVal blnSQLOnly As Boolean = False, Optional ByVal lngMarshalType As Integer = 0 Output: ADODB.Recordset

Table 7-10 (Cont.) SQL Manager Class (clsSQLMgr)

API Call	Method/Property/Object Reference	Description
fValEntitiesForGroup	Method	Return a recordset containing validation entity specifications for the specified validation entity group. Inputs: ByVal strGroupKey As String, Optional ByVal blnReadOnly As Boolean = False, Optional ByVal blnSQLOnly As Boolean = False, Optional ByVal lngMarshalType As Integer = 0 Output: ADODB.Recordset
fValGroup	Method	Return a recordset containing a single validation rule group. Inputs: ByVal strGroupKey As String, Optional ByVal blnReadOnly As Boolean = False, Optional ByVal blnSQLOnly As Boolean = False, Optional ByVal lngMarshalType As Integer = 0 Output: ADODB.Recordset
fValGroupList	Method	Return a recordset containing all validation rule group records. Inputs: Optional ByVal blnIncludeNone As Boolean = False, Optional ByVal blnReadOnly As Boolean = False, Optional ByVal blnSQLOnly As Boolean = False, Optional ByVal lngMarshalType As Integer = 0 Output: ADODB.Recordset

SQL Class (clsSQL)

fdmAPI.API.DataWindow.SQL.API Call

Table 7-11 SQL Class (clsSQL)

API Call	Methods/Properties/Object Reference	Description
fGetAppID	Method	Return the application ID for the Load ID passed in. Inputs: ByVal strLoadID As String Output: String
fGetLanguageCode	Method	Returns the language code. Inputs: ByVal AppID As String Output: String
fGetTopLevelDirectory	Method	Return top level directory. Inputs: ByVal AppID As String Output: String

Table 7-11 (Cont.) SQL Class (clsSQL)

API Call	Methods/Properties/Object Reference	Description
fOptionListForKey	Method	Returns a recordset containing one record with values for the specified option name. Inputs: ByVal strOptionName As String, Optional ByVal blnReadOnly As Boolean = False, Optional ByVal blnSQLOnly As Boolean = False, Optional ByVal lngMarshalType As Integer = 0, Optional ByVal blnTranslateCaptions As Boolean = False Output: ADODB.Recordset

DB Tools Class (clsDBTools)

fdmAPI.API.DataWindow.DBTools.API Call

Table 7-12 DB Tools Class (clsDBTools)

API Call	Methods/Properties/Object Reference	Description
fGlobalOptionExists	Method	Determine when the global system option exists. Inputs: ByVal strOptionName As String Output: Boolean
fGlobalOptionGet	Method	Retrieve a global system option as a string containing the current value of the option. Inputs: ByVal strOptionName As String Output: String
mLog	Method	Write an entry to the database activity log (AIF_PROCESS_LOGS table). Inputs: ByVal intProcessID As Integer, ByVal strEntityType As String, ByVal strEntityName As String, ByVal intLogSequence As Integer, ByVal strLogMessage As String Output: None

Table 7-12 (Cont.) DB Tools Class (clsDBTools)

API Call	Methods/Properties/Object Reference	Description
mLogError	Method	Write an error to a user runtime error log. The log is located in the outbox directory of the application under the file name "LoadID_FDM_API.log". Inputs: ByVal lngErrorCode As Integer, ByVal strErrorDesc As String, ByVal strErrorSource As String Output: None

Maintenance Manager Class (clsMaintenanceMgr)

fdmAPI.API.MaintenanceMgr.API Call

Table 7-13 Maintenance Manager Class (clsMaintenanceMgr)

API Call	Methods/Properties/Object Reference	Description
fProcessStatus	Method	Retrieve the current workflow process status for the specified POV values. The process status is returned as a COM object of the type "objProcessStatus". The best way to determine the current status is to check the lngStatus property of the return object, which contains a value that can be checked against the enumeration "enmProcessState". Inputs: ByVal strLoc As String, ByVal strCat As String, ByVal strPer As String, ByVal strRuleID As String Output: objProcessStatus object objProcessStatus Properties: • strLoc • strCat • strPer • blnImp • strImp • blnTBVal • strTBVal • blnExp • strExp • blnEntLoad • strEntLoad • blnEntVal • strEntVal • lngStatus • dteLastActive • blnCert • strCert • blnAsses • strAsses • blnChildDone • strChildDone • blnUD1 • strUD1 • blnUD2 • strUD2 • blnUD3 • strUD3 • blnUD4 • strUD4 • blnValDirty

Table 7-13 (Cont.) Maintenance Manager Class (clsMaintenanceMgr)

API Call	Methods/Properties/Object Reference	Description
		• blnWCDirty • blnLogicDirty • intLockState

POV Manager Class (clsPOVMgr)

fdmAPI.API.POVMgr.API Call

Table 7-14 POV Manager Class (clsPOVMgr)

API Call	Methods/Properties/Object References	Description
fCategoryKey	Method	Return the numerical category key for the specified category name. Inputs: ByVal strCat As String Output: Integer
fCategoryName	Method	Return the text name of a category for the specified category key. Inputs: ByVal lngCatKey As Integer Output: String
fCheckLocParent	Method	Check when the requested location is assigned as the MAPPING parent to any other location. Inputs: ByVal strLoc As String Output: Boolean

Table 7-14 (Cont.) POV Manager Class (clsPOVMgr)

API Call	Methods/Properties/Object References	Description
fCheckLocProf	Method	Return all behavior attributes for the specified location as a COM object of type "objLocProfile". This object also contains the key information of the location and the data segment assignment. Inputs: ByVal strLoc As String Output: objLocProfile objLocProfile Properties: • blnAdapterImport • blnChild • blnExists • blnScriptImport • blnSeq • lngControlsType • lngLocKey • lngParentLocKey • lngRedFlagLevel • lngSegKey • lngType • strAdaptor • strCLogicGrp • strControlsApprover • strControlsApproverProxy • strControlsGrp1 • strControlsGrp2 • strControlsParent • strCurrency • strDataValue • strImportGrp • strIntgConfig1 • strIntgConfig2 • strIntgConfig3 • strIntgConfig4 • strLogicGrp • strNotes • strParent • strValEntityGrp • strValRuleGrp
fGetLocationName	Method	Return the text name of a location based on its numeric key. Inputs: ByVal lngPartKey As Integer Output: String

Table 7-14 (Cont.) POV Manager Class (clsPOVMgr)

API Call	Methods/Properties/Object References	Description
fPeriodKey	Method	Return attributes associated with the specified period value as a COM object of type "objPeriod". Use this procedure to return the prior period of a location by setting the optional parameter "blnPriorPer" to true. By default the procedure returns the target period associated with the 'M' or 'Monthly' frequency, but the optional parameter "strFreq" can be used to retrieve a target value associated with a different frequency. Inputs: ByVal strPer As String, Optional ByVal blnPriorPer As Boolean = False, Optional ByVal strFreq As String = "" Output: objPeriod objPeriod Properties: • dblSerialKey • dteDateKey • strDateKey • strSerialKey • strTarget • strTargetYear

State Class (clsState)

API State

Table 7-15 State Class (clsState)

API Call	Methods/Properties/Object Reference	Description
AppID	Property	Returns the Application ID for the LoadID
bstMaintenanceMgr	Object Reference	Reference to the clsMaintenanceMgr class
bstSQLMgr	Object Reference	Reference to the clsSQLMgr class
CatKey	Property	Returns the Category Key for the LoadID.
CatName	Property	Returns the Category Name for the LoadID.
DataValue	Property	Returns the DataValue assigned to the Location for the LoadID.
dstConnection	Object Reference	Reference to the clsConnection class
dstDataAccess	Object Reference	Reference to the clsDataAccess class

Table 7-15 (Cont.) State Class (clsState)

API Call	Methods/Properties/Object Reference	Description
dstDataManipulation	Object Reference	Reference to the clsDataManipulation class
dstDBTools	Object Reference	Reference to the clsDBTools class
dstSQL	Object Reference	Reference to the clsSQL class
dstUtilities	Object Reference	Reference to the clsUtilities class
EndPeriodKey	Property	Returns the End Period for the LoadID.
ExportFlag	Property	Returns Y or N. If the Data Rule is executed with Export checked, this property returns Y.
ExportMode	Property	Returns Y or N. If ExportFlag is Y, this property returns the selected mode for Export (Replace or Append). Otherwise, it returns an empty string.
FileDir	Property	Returns the FILE_PATH field from the AIF_BAL_RULE_LOADS table.
FileName	Property	Returns a string containing the name of the file that was imported.
GstrDateMask	Property	Returns the date mask.
ImportFlag	Property	Returns the IMPORT_FROM_SOURCE_FLAG from the AIF_BAL_RULE_LOADS table. If LoadID initiates the Import Process, this flag is set to Y. Otherwise it is set to F.
ImportFormat	Property	Returns the Import Format used for the Load ID being processed.
ImportMode	Property	Returns REPLACE or APPEND based on the LoadID. This property returns null if the LoadID did not Import data (that is, Validate only).
LoadID	Property	Returns the Load ID.
LocKey	Property	Returns the Location Key for the Location being processed for the LoadID.
LocName	Property	Returns the Location Name for the location being processed for the LoadID.
MultiPeriodLoad	Property	Returns True or False. This property is True if multiple periods are being processed. Otherwise, returns False.
PeriodKey	Property	Returns the Period Key for the LoadID.
RuleID	Property	Returns the Rule ID for the Data Rule being processed.
RuleName	Property	Returns the Rule Name for the Data Rule being processed.

Table 7-15 (Cont.) State Class (clsState)

API Call	Methods/Properties/Object Reference	Description
SourceName	Property	Returns the SOURCE_SYSTEM_NAME for the data rule.
SourceType	Property	Returns the SOURCE_SYSTEM_TYPE for the data rule.
TargetAppDB	Property	Returns "NA" for HFM. If the target application type is HPL then this property returns the PLAN_TYPE. If the target application type is ESSBASE this property returns the ESSBASE_DB_NAME.
TargetAppName	Property	Returns the APPLICATION_NAME for the target system.

Visual Basic Sample Scripts

Sample Visual Basic scripts in this section include:

- [Visual Basic Script to Send Email](#)
- [Visual Basic Script to Copy a File to the Inbox](#)
- [Visual Basic Script to Execute SQL](#)
- [Visual Basic Script to Access Various Properties](#)

Visual Basic Script to Send Email

The following script shows how to send an email:

```

*****
'REVIEW INPUT PARAMETERS
*****

If WScript.Arguments.Count <> 5 Then

    Wscript.Echo "You must supply 5 runtime variables." & vbNewLine &
    "Usage: HFM_LOAD.vbs LoadID CSS-Token OracleInstancePath JavaHomePath
    OLE_DB_PROVIDER"

    WScript.Quit -1

Else

    LoadID = Wscript.Arguments.Item(0)

    CSSToken = WScript.Arguments.Item(1)

    OracleInstancePath = WScript.Arguments.Item(2)

```



```

        JavaHomePath = WScript.Arguments.Item(3)

        Provider = WScript.Arguments.Item(4)

    End If

    !*****

    !*****

    'Initialize FDMEE VB API

    !*****

    Set objFDMAPI = CreateObject("fdmapi.clsAppServerDM")

    WScript.Echo "FDMEE VB API created."

    blnInit = objFDMAPI.fInitializeMainAPI(CStr(LoadID), CStr(CSSToken),
    CStr(OracleInstancePath), CStr(JavaHomePath), CStr(Provider))

    If blnInit Then

        WScript.Echo "FDMEE VB API initialized."

    Else

        WScript.Quit -1

        WScript.Echo "fInitializeMainAPI failed."

    End If

    !*****

    WScript.Echo CBool(True)

    !*****

    'Sample Code

    !*****

    If objFDMAPI.API.POVMgr.PPOVLocation = "LOCATION_NAME" Then

        'Check Status of LoadID

        Set objProcessStatus =
objFDMAPI.API.MaintenanceMgr.fProcessStatus(objFDMAPI.API.State.GstrPOVLoc,
objFDMAPI.API.State.GstrPOVCat, objFDMAPI.API.State.GstrPOVPer)

        If objProcessStatus.blnImp Then

            'Import was successful

            'Send an email

            SendEmail("FromUser@domain.com", "ToUser@domain.com",
"Subject", "Import was successful")

        End If
    
```

```
End If

!*****

!*****

'Destroy objects

!*****

objFDMAPI.Dispose

Set objFDMAPI = Nothing

!*****

Sub SendEmail(strFromAddress, strToAddress, strSubject, strEmailMsgLine)

    'Declare local variables

    Dim strFromAddress

    Dim strToAddress

    Dim strSubject

    Dim strEmailMsgLine

    Dim objMsg

    Dim objConfig

    Dim intSendUsing

    Dim strSMTPServer

    Dim intSendPort

    'Initialize Mail Server variables

    'SMTP server name

    strSMTPServer = "YourMailServer"

    'CdoSendUsing enumeration-1=use smtp on local machine, 2=use smtp over network

    intSendUsing = 2

    'SMTP port on server

    intSMTPPort = 25

    'Create CDO configuration object

    Set objConfig = CreateObject("CDO.Configuration")

    'Set Config object settings

    With objConfig.Fields
```

```

        .Item("http://schemas.microsoft.com/cdo/configuration/sendusing") =
intSendUsing

        .Item("http://schemas.microsoft.com/cdo/configuration/smtpserver") =
strSMTPServer

        .Item("http://schemas.microsoft.com/cdo/configuration/smtpserverport")=
intSMTPPort

        .Item("http://schemas.microsoft.com/cdo/configuration/smtpconnectiontimeout") =
10

        .Update

End With

'Create CDO message object
Set objMsg = CreateObject("CDO.Message")

'Assign config object to configuration property of message object
Set objMsg.Configuration = objConfig

'Set Message object settings and send mail
With objMsg

    .To = strToAddress

    .From = strFromAddress

    .Subject = strSubject

    .TextBody = strEmailMsgLine

    .Send

End With

'Destroy message objects
Set objMsg = Nothing

Set objConfig = Nothing

End Sub

```

Visual Basic Script to Copy a File to the Inbox

The following script shows how to copy a file to the Inbox:

```

*****

'REVIEW INPUT PARAMETERS

!*****

If WScript.Arguments.Count <> 5 Then

```

```
Wscript.Echo "You must supply 5 runtime variables." & vbNewLine & "Usage:
HFM_LOAD.vbs LoadID CSS_Token OracleInstancePath JavaHomePath OLE_DB_PROVIDER"

WScript.Quit -1

Else

    LoadID = Wscript.Arguments.Item(0)

    CSSToken = WScript.Arguments.Item(1)

    OracleInstancePath = WScript.Arguments.Item(2)

    JavaHomePath = WScript.Arguments.Item(3)

    Provider = WScript.Arguments.Item(4)

End If

'*****

'*****

'Initialize FDMEE VB API

'*****

Set objFDMAPI = CreateObject("fdmapi.clsAppServerDM")

WScript.Echo "FDMEE VB API created."

blnInit = objFDMAPI.fInitializeMainAPI(CStr(LoadID), CStr(CSSToken),
CStr(OracleInstancePath), CStr(JavaHomePath), CStr(Provider))

If blnInit Then

    WScript.Echo "FDMEE VB API initialized."

Else

    WScript.Quit -1

    WScript.Echo "fInitializeMainAPI failed."

End If

'*****

'*****

'Sample Code

'*****

If objFDMAPI.API.State.LocName = "MT_COMMA7" Then

'*****

    'Copy file to Inbox
```



```

*****

    strFileDir = "C:\"

    strFileName = "CopyThisFile.txt"

    'Create FileSystemObject

    Set objFSO = CreateObject("Scripting.FileSystemObject")

    If objFSO.FileExists(strFileDir & strFileName) Then

        'Copy the file

    End If

    Set objFSO = Nothing

End If

! *****

```

Visual Basic Script to Execute SQL

The following script shows how to execute SQL:

```

! *****

'REVIEW INPUT PARAMETERS

! *****

If WScript.Arguments.Count <> 5 Then

    Wscript.Echo "You must supply 5 runtime variables." & vbNewLine & "Usage:
HFM_LOAD.vbs LoadID CSS_Token OracleInstancePath JavaHomePath OLE_DB_PROVIDER"

    WScript.Quit -1

Else

    LoadID = Wscript.Arguments.Item(0)

    CSSToken = WScript.Arguments.Item(1)

    OracleInstancePath = WScript.Arguments.Item(2)

    JavaHomePath = WScript.Arguments.Item(3)

    Provider = WScript.Arguments.Item(4)

End If

! *****

! *****

'Initialize FMEE VB API

! *****

```

```
Set objFDMAPI = CreateObject("fdmapi.clsAppServerDM")

WScript.Echo "FDMEE VB API created."

blnInit = objFDMAPI.fInitializeMainAPI(CStr(LoadID), CStr(CSSToken),
CStr(OracleInstancePath), CStr(JavaHomePath), CStr(Provider))

If blnInit Then

    WScript.Echo "FDMEE VB API initialized."

Else

    WScript.Quit -1

    WScript.Echo "fInitializeMainAPI failed."

End If

! *****

! *****

'Sample Code

! *****

If objFDMAPI.API.State.LocName = "MT_COMMA7" Then

! *****

    'Delete one record from tdataseg

! *****

    strDeleteSQL = "DELETE FROM TDATESEG_T WHERE LOADID = " & CStr(LoadID) & "
AND ENTITY = 'EastSales'"

    'Execute SQL

    objFDMAPI.API.DataWindow.DataManipulation.fExecutedML CStr(strDeleteSQL)

End If

! *****

! *****

'Destroy objects

! *****

objFDMAPI.Dispose

Set objFDMAPI = Nothing

WScript.Echo "FDMEE VB API destroyed."
```

Visual Basic Script to Access Various Properties

The following script shows how to access various properties:

```

*****

Visual Basic Script to Access Various Properties

The following script shows how to access various properties:

*****

'REVIEW INPUT PARAMETERS

*****

If WScript.Arguments.Count <> 5 Then

    Wscript.Echo "You must supply 5 runtime variables." & vbNewLine & "Usage:
HFM_LOAD.vbs LoadID CSS_Token OracleInstancePath JavaHomePath OLE_DB_PROVIDER"

    WScript.Quit -1

Else

    LoadID = Wscript.Arguments.Item(0)

    CSSToken = WScript.Arguments.Item(1)

    OracleInstancePath = WScript.Arguments.Item(2)

    JavaHomePath = WScript.Arguments.Item(3)

    Provider = WScript.Arguments.Item(4)

End If

*****

*****

'Initialize FDMEE VB API

*****

Set objFDMAPI = CreateObject("fdmapi.clsAppServerDM")

WScript.Echo "FDMEE VB API created."

blnInit = objFDMAPI.fInitializeMainAPI(CStr(LoadID), CStr(CSSToken),
CStr(OracleInstancePath), CStr(JavaHomePath), CStr(Provider))

If blnInit Then

    WScript.Echo "FDMEE VB API initialized."

Else

    WScript.Quit -1

    WScript.Echo "fInitializeMainAPI failed."

End If

*****

```

```
*****

'Sample Code

*****

WScript.Echo "AppID = " & objFDMAPI.API.State.AppID
WScript.Echo "CatKey = " & objFDMAPI.API.State.CatKey
WScript.Echo "CatName = " & objFDMAPI.API.State.CatName
WScript.Echo "DataValue = " & objFDMAPI.API.State.DataValue
WScript.Echo "EndPeriodKey = " & objFDMAPI.API.State.EndPeriodKey

'Need to test this during Export process

WScript.Echo "ExportFlag = " & objFDMAPI.API.State.ExportFlag
WScript.Echo "ExportMode = " & objFDMAPI.API.State.ExportMode
WScript.Echo "FileDir = " & objFDMAPI.API.State.FileDir
WScript.Echo "FileName = " & objFDMAPI.API.State.FileName
WScript.Echo "GblnPOVLocal = " & objFDMAPI.API.State.GblnPOVLocal
WScript.Echo "GdtePOVPerKey = " & objFDMAPI.API.State.GdtePOVPerKey
WScript.Echo "GlngPOVCatKey = " & objFDMAPI.API.State.GlngPOVCatKey
WScript.Echo "GlngPOVLocKey = " & objFDMAPI.API.State.GlngPOVLocKey
WScript.Echo "GlngPOVLocType = " & objFDMAPI.API.State.GlngPOVLocType
WScript.Echo "GstrDateMask = " & objFDMAPI.API.State.GstrDateMask
WScript.Echo "GstrMethodID = " & objFDMAPI.API.State.GstrMethodID
WScript.Echo "GstrMsg = " & objFDMAPI.API.State.GstrMsg
WScript.Echo "GstrPOVCat = " & objFDMAPI.API.State.GstrPOVCat
WScript.Echo "GstrPOVLoc = " & objFDMAPI.API.State.GstrPOVLoc
WScript.Echo "GstrPOVPer = " & objFDMAPI.API.State.GstrPOVPer
WScript.Echo "ImportFlag = " & objFDMAPI.API.State.ImportFlag
WScript.Echo "ImportFormat = " & objFDMAPI.API.State.ImportFormat
WScript.Echo "ImportMode = " & objFDMAPI.API.State.ImportMode
WScript.Echo "LoadID = " & objFDMAPI.API.State.LoadID
WScript.Echo "LocKey = " & objFDMAPI.API.State.LocKey
WScript.Echo "LocName = " & objFDMAPI.API.State.LocName
WScript.Echo "MultiPeriodLoad = " & objFDMAPI.API.State.MultiPeriodLoad
```

```

WScript.Echo "PeriodKey = " & objFDMAPI.API.State.PeriodKey

WScript.Echo "RuleID = " & objFDMAPI.API.State.RuleID

WScript.Echo "RuleName = " & objFDMAPI.API.State.RuleName

WScript.Echo "Scenario = " & objFDMAPI.API.State.Scenario

WScript.Echo "SourceName = " & objFDMAPI.API.State.SourceName

WScript.Echo "SourceType = " & objFDMAPI.API.State.SourceType

WScript.Echo "TargetAppDB = " & objFDMAPI.API.State.TargetAppDB

WScript.Echo "TargetAppName = " & objFDMAPI.API.State.TargetAppName

If objFDMAPI.API.POVMgr.PPOVLocation = "LOCATION_NAME" Then

    'Enter Sample Code HERE

Else

    'Enter Sample Code HERE

End IF

! *****

! *****

'Destroy objects

! *****

objFDMAPI.Dispose

Set objFDMAPI = Nothing

! *****

```

8

FDMEE Reports

Oracle Hyperion Financial Data Quality Management, Enterprise Edition provides prebuilt reporting tools that capture business-critical operations and revenue-generating activities within your organization. These reports provide key information on how metadata and data are integrated from the source to the target.

The FDMEE reporting framework represents a unified solution that incorporates source and target data, templates, and user-defined SQL queries. Templates, created in Oracle Business Intelligence Publisher, consume data in XML format and generate reports dynamically. You can add SQL queries to extract data from tables, or couple them with the report parameters to extend the definition of a standard report.

The FDMEE reporting framework has the following components:

FDMEE reports can be generated as PDF, Excel, Word, or HTML output.

- Query Definition
- Report Group
- Report Definition
- Report Execution

FDMEE Reports

The standard Oracle Hyperion Financial Data Quality Management, Enterprise Edition report groups are described below. For information on the subcategories of each report, see [FDMEE Detail Reports](#).

Base Trial Balance (With Rules)

Location: KS7DIM_FILE
Category: Actual
Period: 2005-01-01
Currency: [NONE]

GL Account	GL Center	GL Account Description	Amount	Acct Rule	Entity Rule
Travel	EastAdmin		2,991.37	ACCT_DEFAULT	ACCT_DEFAULT
Travel	WestAdmin		1,937.34	ACCT_DEFAULT	ACCT_DEFAULT
Travel	EastSales		381.33	ACCT_DEFAULT	ACCT_DEFAULT
Travel	WestSales		1,368.48	ACCT_DEFAULT	ACCT_DEFAULT
		Total	6,678.51		
Software	EastAdmin		1,197.38	ACCT_DEFAULT	ACCT_DEFAULT
Software	EastSales		152.64	ACCT_DEFAULT	ACCT_DEFAULT
Software	WestSales		547.77	ACCT_DEFAULT	ACCT_DEFAULT
Software	WestAdmin		775.47	ACCT_DEFAULT	ACCT_DEFAULT
		Total	2,673.26		
Meals	EastAdmin		1,000.06	ACCT_DEFAULT	ACCT_DEFAULT
Meals	WestSales		457.62	ACCT_DEFAULT	ACCT_DEFAULT
Meals	WestAdmin		648.26	ACCT_DEFAULT	ACCT_DEFAULT
Meals	EastSales		127.60	ACCT_DEFAULT	ACCT_DEFAULT
		Total	2,234.74		
BuildingDepr	EastAdmin		1,509.86	ACCT_DEFAULT	ACCT_DEFAULT
BuildingDepr	WestSales		1,600.00	ACCT_DEFAULT	ACCT_DEFAULT
BuildingDepr	EastSales		3,838.17	ACCT_DEFAULT	ACCT_DEFAULT
BuildingDepr	WestAdmin		183.32	ACCT_DEFAULT	ACCT_DEFAULT
		Total	7,131.35		
Advertising	EastAdmin		1,281.62	ACCT_DEFAULT	ACCT_DEFAULT
Advertising	WestSales		577.16	ACCT_DEFAULT	ACCT_DEFAULT
Advertising	EastSales		180.82	ACCT_DEFAULT	ACCT_DEFAULT
Advertising	WestAdmin		817.08	ACCT_DEFAULT	ACCT_DEFAULT
		Total	2,816.68		
TransportationDepr	EastSales		1,584.40	ACCT_DEFAULT	ACCT_DEFAULT
TransportationDepr	WestAdmin		75.67	ACCT_DEFAULT	ACCT_DEFAULT
TransportationDepr	EastAdmin		623.27	ACCT_DEFAULT	ACCT_DEFAULT
TransportationDepr	WestSales		575.00	ACCT_DEFAULT	ACCT_DEFAULT

12/8/2013 12:44:20 PM

Page 1 of 37

Working with Query Definitions

Two types of SQL queries can be used in a report definition: a base query and a parameter query.

A base query enables users to pull data from various tables and display the data as report output. The base extends the definition of a standard report, and can be reused with multiple reports. For example, using one query definition, you can show different columns or groupings. In one report, you can list amounts by account and group by entity, and in another list amount by entity and group by account.

The parameter SQL query enables you to run a query against the parameters in the report definition. For example, use the query to select the Location, Period, Category, or Account.

For information on the tables and joins you can use to build your SQL queries, see [TDATASEG Table Reference](#) and [TLOGPROCESS Table Reference](#) in Appendix G. The TDATASEG table is used to store the data loaded by the user, the transformation between the source dimension members, and the results of the mapping process. The TLOGPROCESS table is used to store the workflow process status for a location, category, and period.

You can save a query definition as an XML file, which, in turn, you can use to create custom templates using Oracle Business Intelligence Publisher or the BI Publisher desktop addin for Microsoft Word.

To add a query definition:

1. On the **Setup** tab, under **Reports**, select **Query Definition**.
2. In **Query Definition**, click **Add**.
3. In **Name**, enter the name of the query definition.

Oracle recommends that you assign a name that corresponds to the report definition in which the SQL is embedded.

4. In **Select Clause**, specify the SQL Select clause used to query the database and return the data that corresponds to your selected criteria.

5. In **Where Clause**, specify the SQL Where clause used to restrict the data that is returned to only the specific criteria that you specify.

6. In **Group by/Order by Clause**, specify the Group by or Order by clause.

The ORDER BY clause sorts the records in the result set. The ORDER BY clause can be used only in SQL SELECT statements.

The GROUP BY clause fetches data across multiple records and returns the results grouped by one or more columns.

7. Click **Validate Query**.

If the query definition is validated, Oracle Hyperion Financial Data Quality Management, Enterprise Edition returns the message: "Query validation successful."

If the query definition is not validated, FDMEE indicates that an error was found in the SQL. You must fix the error before validating the query again.

8. Click **Save**.

9. **Optional:** To save the query definition to an XML file, click **Generate XML**.

Working with Report Definitions

Report definitions are the attributes that determine the content and structure of a report. Using the Report Definition option, you can:

- Associate a report to a Report Group.
- Associate a query definition with the report.
- Associate a Rich Text Format template (RTF) with the report.
- Define report parameters.

Adding Report Groups

Report groups enable you to assemble similar types of reports into one category for ease of use.

To add a report group:

1. On the **Setup** tab, under **Reports**, select **Report Definition**.
2. In **Report Definition**, select the **Report Group** tab.
3. In **Report Group**, click **Add**.

4. In the blank **Name** field, enter the title of the report group.

For example, enter "Base Trial Balance Reports."

5. In **Sequence**, enter a numeric value that identifies the display order of the report group on the Report Definition screen.
6. Click **Save**.

Working with Report Definitions

A report definition can include a query definition created in SQL and any RTF template set up for use with a selected report. Additionally, you can associate a report with the query definition that prompts the user with a list of values when selecting the parameters of the report.

To add a report definition:

1. On the **Setup** tab, under **Reports**, select **Report Definition**.
2. In **Report Definition**, select the **Reports** tab.

The Report tab consists of three regions:

- **Summary**—Lists all report definitions.
 - **Details**—Shows the report name, associated base query definition, report group, and associated template.
 - **Parameters**—Shows the parameter name and type, display name, sequence, parameter value, and any query definition used to supply a list of values for a given report parameter.
3. In the summary grid, click **Add**.
 4. In the detail grid, in **Name**, enter the name of the report definition.
 5. In **Group**, select the report group name associated with the definition.

To search on a report group, click  and choose a report group from the Search and Select: Group screen.

Report groups are created on the Report Group tab. See [Adding Report Groups](#).

6. In **Query**, select the name of the query definition to associate with the report definition.

To search on a query definition, click  and choose a query definition from the Search and Select: Query screen.

Query definitions are defined in the Query Definition option. See [Working with Query Definitions](#).

7. In **Template**, select the RTF template to associate with the report definition.

To search on a template, click  and choose a template from the Search and Select: Template screen.

Oracle Hyperion Financial Data Quality Management, Enterprise Edition report templates are saved with a Rich Text Format (RTF) file type. The RTF file type is a specification used by common word processing applications, such as Microsoft Word. The templates available to the FDMEE report are created in Oracle Business Intelligence Publisher. See the *Report Designer's Guide for Oracle Business Intelligence Publisher*. To download the Oracle BI Publisher Desktop for Microsoft Office, see: [Oracle BI Publisher Downloads](#).

 Note:

You can view report descriptions and parameters in the language selected for your user interface. Language templates are available at: \oracle\middleware\epmsystem11R1\products\FinancialDataQuality\templates for you to select and associate with a report definition.



To upload a template, click . In **Search and Select: Template**, click **Upload**. Then click **Browse** to navigate to the template to upload and click **OK**.

8. Click **Save**.

To define the parameters of the report definition:

1. From the summary grid, select the name of the report definition to which to add parameters.

2. In the detail grid, click **Add**.

A blank line is displayed to which to add report parameters.

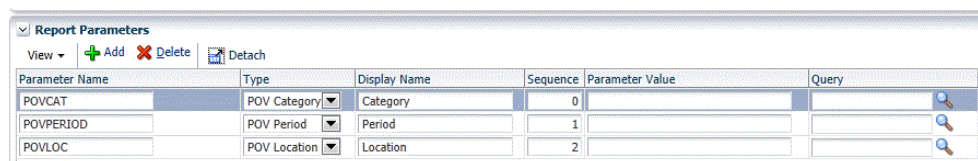
3. In **Parameter Name**, enter the POV category to which the parameter belongs.

For example, enter POVL0C, for the POV location, POVPeriod for the POV period, POVCAT for POV category, and POVTARGET for the target category.

4. In **Type**, select the type of parameter for the report.

Available types:

- **POV**—Prompts for the POV location, period, category or rule. The POV values default from the users POV when they run the report parameter.
- **Query**—The Query type enables you to create a query that populates a list of values that a user can select from when running the report. The parameter name entered by the user must match a parameter used in the assigned query. For example, if the parameter name is “California,” a corresponding parameter in the query must be named California. The name in the query must have a “~” character as a prefix so that the system can parse the query correctly.
- **Static**—A Static type Indicates that the parameter is a constant or a predefined value, and the parameter value is supplied by the user. When the parameter is supplied and the report is executed, the query uses this parameter as part of the Select statement.
- **Title** (indicates the header part of column title)



Parameter Name	Type	Display Name	Sequence	Parameter Value	Query
POVCAT	POV Category	Category	0		
POVPERIOD	POV Period	Period	1		
POVL0C	POV Location	Location	2		

5. In **Display Name**, enter the name to display for the parameter on the Generate Report screen.

6. In **Parameter Value**, enter the value for the "Static" parameter type. For example, if you have an aging report for 30 days, enter 30 in this field.
7. In **Sequence**, enter the display order of the parameter on the Generate Report screen.
8. In **Query**, specify the query definition that provides a list of values for the parameter. For example, to show a list of data rules for the user to select in the Generate Report screen, specify the query definition to define these queries.
9. Click **Save**.

To copy a report:

1. On the **Setup** tab, under **Reports**, select **Report Definition**.
2. In **Report Definition**, in the **Report** summary grid, select the report.
3. In the **Report** summary grid, click **Copy Current Report**.

The copied report is added to the list of reports. The name of the report takes the original report name appended with "_copy."

Creating a Report Template

Report templates, created in Oracle Business Intelligence Publisher, consume data in XML format and generate reports dynamically.

The screenshot shows the Oracle Business Intelligence Publisher interface. At the top, there is a 'Summary' tab with buttons for 'View', 'Add', 'Delete', 'Detach', 'Validate Query', and 'Generate XML'. Below this, a list of reports is displayed. The 'Resulting XML' section shows the following XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xmlIVO>
  <xmlVORow>
    <PARTITIONKEY>DEMO-LOC</PARTITIONKEY>
    <TARGACCTINDEX>1</TARGACCTINDEX>
    <TARGACCTGROUP>ALL</TARGACCTGROUP>
    <TARGACCTDESC>NA</TARGACCTDESC>
    <TARGACCTTYPE>NA</TARGACCTTYPE>
    <SRCACCTKEYX>Accounts Payable</SRCACCTKEYX>
    <SRCACCTKEY>2210</SRCACCTKEY>
    <SRCCENTERKEY>01</SRCCENTERKEY>
    <SRCACCTDESC null="true" />
    <CHANGESIGN>0</CHANGESIGN>
    <SRCAMOUNT>-562.5</SRCAMOUNT>
    <CATNAME>Current</CATNAME>
    <PERIODKEY>2013-01-01</PERIODKEY>
    <CATKEY>1</CATKEY>
    <PARTCURRENCYKEY>Local</PARTCURRENCYKEY>
  </xmlVORow>
</xmlIVO>
```

The 'Template Builder' section shows a tree view of the XML structure. The 'Field' dialog box is open, showing the 'Find' field and the 'Find Next' button. The 'Example' section shows the 'Calculation' field and the 'On Grouping' checkbox. The 'Force LTR Direction' checkbox is also present. The 'Insert', 'Close', and 'Help' buttons are at the bottom. The 'Please select a node in the tree.' message is displayed.

To create a report template:

1. Create a query definition for the report.

- a. On the **Setup** tab, under **Reports**, select **Query Definition**.
 - b. In **Query Definition**, click **Add**.
 - c. In **Name**, enter the name of the query definition.
Oracle recommends that you assign a name that corresponds to the report definition in which the SQL is embedded.
 - d. In **Select Clause**, specify the SQL Select clause used to query the database and return the data that corresponds to your selected criteria.
 - e. In **Where Clause**, specify the SQL Where clause used to restrict the data that is returned to only the specific criteria that you specify.
 - f. In **Group by/Order by Clause**, specify the Group by or Order by clause.
The ORDER BY clause sorts the records in the result set. The ORDER BY clause can be used only in SQL SELECT statements.
The GROUP BY clause fetches data across multiple records and returns the results grouped by one or more columns.
 - g. Click **Validate Query**.
If the query definition is validated, Oracle Hyperion Financial Data Quality Management, Enterprise Edition returns the message: "Query validation successful."
If the query definition is not validated, FDMEE indicates that an error was found in the SQL. You must fix the error before validating the query again.
 - h. Click **Save**.
2. Click **Generate XML**.
 3. Create the report template.
 - a. Install the XML Publisher Desktop.
See the *Report Designer's Guide for Oracle Business Intelligence Publisher*. To download the Oracle BI Publisher Desktop for Microsoft Office, see [Oracle BI Publisher Downloads](#).
 - b. Load the XML created in step 2.
Reference the template builder tutorial as needed.
 - c. Save the template, and then upload it to the server.
 4. Create report definition with group, query and template.

Creating a XLIFF File for Translated Report Templates

When you required to have a translation of the text strings of the template layout, you can generate an XLIFF (or RTF) file from the report template.

A "translatable string" is any text in the template that is intended for display in the published report, such as table headers and field labels. Text supplied at runtime from the data is not translatable, nor is any text that you supply in the Microsoft Word form fields. You can translate the template XLIFF file into as many languages as desired and then associate these translations to the original template.

XLIFF is the XML Localization Interchange File Format. It is the standard format used by localization providers. For more information about the XLIFF specification, see: [XLIFF 1.1 Specification](#).

To create an XLIFF translation file:

1. Open your template in **Microsoft Word with the Template Builder for Word** installed.
2. From the **Template Builder** menu, select **Tools**, then **Translations**, and then **Extract Text**.
BI Publisher extracts the translatable strings from the template and exports them to an XLIFF (.xlf) file.
3. When prompted, save the file as: **TemplateName_<language code>.xlf or .rtf** where:
 - **TemplateName** is the original template name.
 - **language code** is the two-letter ISO language code (use lower case characters).

For example, if your original template is named EmployeeTemplate and you are uploading a translation for Japanese-Japan, name the file: EmployeeTemplate_ja.xlf.

For information on including the "territory code" in the file name, see [Oracle Business Intelligence Publisher Report Designer's Guide](#).

4. In the report **Editor**, select the **Layouts** page to upload the translated XLIFF files.
5. Save the file to the language subdirectory of the **<EPM_ORACLE_HOME>\Products\FinancialDataQuality\Templates** folder.

Running Reports

To run reports:

1. On the **Workflow** tab, under **Other**, select **Report Execution**.
2. In **Report Execution**, in **Report Groups**, select a report group.
3. In **Reports**, select a report.

To filter the display listing by a report name within a report group, enter the name of the report in the blank entry line above the **Name** field and press Enter. For example, to view only reports beginning with **Account**, enter **Account** and press Enter.

To filter the display listing by a base query name within a report group, enter the query name in the blank entry line above **Query**.

4. Click **Execute**.
5. When prompted, enter parameter values on the **Generate Report** screen.
 - a. If applicable, modify the **Period**, **Category**, and **Location** values.
 - b. From the **Report Output Format**, select the output format.

Available output formats are:

- PDF

- HTML
 - EXCEL (.XLS)
- c. From **Execution Mode**, select the online method of running the report.
The online method processes the report immediately.
 - d. **Optional:** To create a report batch script that generates a report when the scripts are executed, click **Create Report Script**.
 - e. Click **OK**.

To create a report script:

1. On the **Workflow** tab, under **Other**, select **Report Execution**.
2. In **Report Execution**, in **Report Groups**, select a report group.
3. In **Reports**, select a report.

To filter the display listing by a report name within a report group, enter the name of the report in the blank entry line above the **Name** field and press Enter. For example, to view only reports beginning with **Account**, enter **Account** and press Enter.

To filter the display listing by a base query name within a report group, enter the query name in the blank entry line above **Query**.

4. Click **Create Report Script**.



Note:

Make sure that password encryption has been set up before creating a report script. See [Using Password Encryption](#).

5. On the **Generate Report Script** screen, select the parameter values.
6. In **Report Output Format**, select the output format of the batch.
7. Click **OK**.

A Report Batch Script window is displayed showing the parameters of the reports script.

Report Batch Script

```
runreport.bat <username> <password> "Check Report
Period Range (Cat, Start Per, End Per)" "Category=Actual"
"Start Period=Mar-05" "Location=KSHPC1_KSAPP" "Report
Output Format=PDF"
```

Scripts are located in the EPM Oracle Instance/FinancialDataQuality directory.

8. Copy the script that Oracle Hyperion Financial Data Quality Management, Enterprise Edition generates, and on a Windows command window or UNIX shell, paste the script, and then run it.

For example at a Windows command line, specify:

```
runreport.bat <username><password> "Check Report With Warnings"  
"Category=Actual" "Period=March-05" "Location=PKA_TEST" "Report Output  
Format=PDF"
```

 Note:

When passing program arguments for a batch file execution, Jython removes double quotes (") unless arguments have a leading space in them. Jython uses double quotes for escaping. To avoid conflicts, add a leading space in the argument. For example, instead of passing "Period=Mar-2003", pass " Period=Mar-2003".

 Note:

You can use a file with an encrypted password when executing reports from batch report scripts, for example, runreport.bat <username><password> file...

FDMEE Detail Reports

The following reports are available in Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

Audit Reports

An audit report displays all transactions for all locations that compose the balance of a target account. The data returned in this report depends on the location security assigned to the user.

Account Chase Wildcard (TargAcct, Per, Cat)

Shows imported accounts for all Oracle Hyperion Financial Data Quality Management, Enterprise Edition locations, subtotaed by FDMEE location, based on an account selection that enables use of wildcards.

Runs for

All FDMEE locations

Parameters

Target account, Period, Category

Query

Account Chase Wildcard

Template

Account Chase WildCard.rtf

Account Chase - Freeform (TargAcct, Per, Cat)

Shows one imported account for all Oracle Hyperion Financial Data Quality Management, Enterprise Edition locations; subtotaled by FDMEE location.

Runs for

All FDMEE locations

Parameters

Target account, Period, Category

Query

Account Chase Freeform

Template

Account Chase Free Form.rtf

Map Monitor for Location

Displays a list of mapping changes, additions, and deletions made for a location based on a date range. Shows the user name, and the time and date of the change.

 Note:

The Map Monitor reports do not capture historical data earlier than release 11.1.2.4.100.

Map Monitor reports are enabled only if the Enable Map Audit is set to "Yes" in System Settings.

Map Monitor reports includes data load mappings from:

- Data load mapping option
- Text file imports
- Lifecycle Management imports
- Oracle Data Relationship Management

Runs for

All Oracle Hyperion Financial Data Quality Management, Enterprise Edition locations

Parameters

Location, Start Date, End Date

Query

Dimension Map Query

Template

Dimension Map for POV.rtf

Map Monitor for User

Displays a list of mapping changes, additions, and deletions made by a user based on a date range. Report shows the user name, and the time and date of the change.



Note:

The Map Monitor reports do not capture historical data earlier than release 11.1.2.4.100.

Map Monitor reports are enabled only if the Enable Map Audit is set to "Yes" in System Settings.

Map Monitor reports includes data load mappings from:

- Data load mapping option
- Text file imports
- Oracle Hyperion Enterprise Performance Management System Lifecycle Management imports
- Oracle Data Relationship Management

Runs for

All Oracle Hyperion Financial Data Quality Management, Enterprise Edition locations

Parameters

User name, Start Date, End Date

Query

Dimension Map for POV

Template

Dimension Map for POV.rtf

Intersection Drill Down (Per, Cat)

Shows target accounts and amounts; and includes drill-down list of source accounts and amounts that are mapped to target accounts.

Runs for

Current Oracle Hyperion Financial Data Quality Management, Enterprise Edition location

Parameters

Period, Category

Query

Intersection Drill Down Query

Template

Intersection Drill Down.rtf

Check Reports

Check reports provide information on the issues encountered when data load rules are run. Note that Check reports return target system values that include aggregation or calculations from the target system.

Note the following when using check reports:

- When the check report is run and opened from the Workbench, it is not saved to the Oracle Hyperion Financial Data Quality Management, Enterprise Edition folder on the server.
- When you run a data rule, a check rule report is not generated automatically. In this case, run the data rule before executing the check report.
- If you run the report in offline mode, the report is saved to the outbox on the FDMEE server.
- To run a data rule and report in batch mode, run the data load rule from a BAT file, and then the report from a BAT file. In this case, you can put each in the same BAT file, or call each of them from a BAT file.

Check Report

Shows the results of the validation rules for the current location (indicates pass or fail status).

Runs for

Current Oracle Hyperion Financial Data Quality Management, Enterprise Edition location

Parameters

Period, Location and Category

Query

Check Report

Template

Check Report.rtf

Check Report Period Range (Cat, Start Per, End Per)

Shows the results of the validation rules for selected periods.

Runs for

Current Oracle Hyperion Financial Data Quality Management, Enterprise Edition
location

Parameters

Category, Start Period, End Period

Query

Check Report Within Period Query

Template

Check Report With Period Range.rtf

Check Report With Warnings

Shows the results of the validation rules for the current location (warnings are recorded in validation rules).

Runs for

Current Oracle Hyperion Financial Data Quality Management, Enterprise Edition
location

Parameters

None

Query

Check Report With Warning

Template

Check Report With Warning.rtf

Check Report By Validation Entity Seq.

Shows the results of the validation rules for the current location (indicates pass or fail status); sorted by the sequence defined in the validation entity group.

Runs for

Current Oracle Hyperion Financial Data Quality Management, Enterprise Edition
location

Parameters

None

Query

Check Report By Validation Entity

Template

Check Report By Validation Entity Sequence.rtf

Base Trial Balance Reports

The Trial Balance reports provide detail on how source data is processed in Oracle Hyperion Financial Data Quality Management, Enterprise Edition. Typically, the Trial Balance is used to display account balances in the General ledger system. As data is loaded from the source General Ledger system to the target EPM application, you can validate and compare the balances loaded with the source Trial Balance amounts.

 Note:

Before running the base Trial Balance Reports, confirm that the user who runs the base Trial Balance reports has access to the location associated with the report. (See [Defining Location Security](#)).

TB Current Location, with Targets (Cat, Per)

Shows imported source accounts (departments) and their corresponding accounts (entities).

Runs for

Current Oracle Hyperion Financial Data Quality Management, Enterprise Edition location

Parameters

Category, Period

Query

Current Trial Balance With Location with Targets

Template

TB Location With Targets.rtf

TB Current Location with Rules (Cat, Per)

Shows imported source accounts (departments) and the mapping entity rule (map wildcard) in which the accounts are included.

Runs for

Current Oracle Hyperion Financial Data Quality Management, Enterprise Edition
location

Parameters

Category, Period

Query

TB Location With Query

Template

TB Location with Rules.rtf

TB Current Locations, All Dimensions-Targets, by Target Entity-Account (Cat, Per)

Shows all imported records with all dimensions and their respective targets: grouped by target entity and account.

Runs for

Current Oracle Hyperion Financial Data Quality Management, Enterprise Edition
location

Parameters

Category, Period

Query

Trial Balance Current Location with Targets

Template

TB/(All Dimensions with Targets) by Target Entity Account.rtf

TB Current Locations, All Dimensions-Targets (Cat, Per)

Shows all imported records with all dimensions and their respective targets.

Runs for

Current Oracle Hyperion Financial Data Quality Management, Enterprise Edition
location

Parameters

Category, Period

Query

Trial Balance Location All Dimension.

Template

TB with Transaction Currency.rtf

TB Current Location, by Target Acct (Cat, Per)

Shows imported accounts: subtotaled by target accounts.

Runs for

Current Oracle Hyperion Financial Data Quality Management, Enterprise Edition
location

Parameters

Category, Period

Query

Trial Balance Current Location Sorted By Target Account

Template

TB With Target Account.rtf

TB Current Location, By Target Entity Account (Cat, Per)

Shows all imported records with all dimensions and their respective targets; grouped
by target entity and account.

Runs for

Current Oracle Hyperion Financial Data Quality Management, Enterprise Edition
location

Parameters

Category, Period

Query

Trial Balance Base Transaction Currency

Template

Base Trial Balance (All Dimensions with Targets).rtf

TB Converted Current Location by Target Entity/Account

Shows imported accounts and entities in addition to original and converted accounts:
subtotaled by target entity.

Runs for

Current Oracle Hyperion Financial Data Quality Management, Enterprise Edition
location

Parameters

Category, Period

Query

Trial Balance Converted by Target Entity/Account Query

Template

TB Converted Current Location by Target Entity Account.rtf

Listing Reports

Listing reports summarize metadata and settings (such as the import format, or check rule) by the current location.

Import Formats by Location

Displays a list of all import formats; sorted by Oracle Hyperion Financial Data Quality Management, Enterprise Edition location.

Runs for

N/A

Parameters

None

Query

Import Format By Location

Template

Import Format by Location.rtf

Location Listing

Shows a list of all mapping rules for a selected period, category, or dimension.

Runs for

Current Oracle Hyperion Financial Data Quality Management, Enterprise Edition location

Parameters

Any FDMEET Dimension, Period, Category

Query

Location Listing Query

Template

Location Listing.rtf

Location Analysis

Location Analysis reports provide dimension mapping by the current location.

Dimension Map (Dimension)

Displays a list of all mapping rules for a selected dimension.

Runs for

Current Oracle Hyperion Financial Data Quality Management, Enterprise Edition location

Parameters

Current FDMEE dimension

Query

Dimension Map

Template

Dimension Map.rtf

Dimension Map For POV (Dimension, Cat, Per)

Shows a list of all mapping rules for a selected period, category, or dimension.

Runs for

Current Oracle Hyperion Financial Data Quality Management, Enterprise Edition location

Parameters

Any FDMEE Dimension, Period, Category

Query

Dimension Map for POV

Template

Dimension Map.rtf

Process Monitor Reports

The Process Monitor Reports shows locations and their positions within the data conversion process. You can use the process monitor report to monitor the status of the closing process. The report is time-stamped. Therefore, it can be used to determine to which locations at which time data was loaded.

Process Monitor (Cat, Per)

Shows all locations and their current status (import, validate, export, load, or check).
(Locations are displayed alphabetically.)

Runs for

All Oracle Hyperion Financial Data Quality Management, Enterprise Edition locations

Parameters

Category, Period

Query

Process Monitor

Template

Process Monitor.rtf

Process Status Period Range (Cat, Start Per, End Per)

Shows a list of all locations and the current load status of each location for each period
of a period range.

Runs for

All Oracle Hyperion Financial Data Quality Management, Enterprise Edition locations,
period range

Parameters

Category, Start Period, End Period

Query

PMPeriodRange

Template

PMPeriodRange.rtf

Process Monitor All Categories (Cat, Per)

Shows a list of all locations and the current load status of every location for every
category.

Runs for

All Oracle Hyperion Financial Data Quality Management, Enterprise Edition categories
and locations

Parameters

Period

source

Query

Process Monitor All Categories

Template

Process Monitor All Category.rtf

Variance Reports

The Variance reports display source and trial balance accounts for one target account, showing data over two periods or categories.

Account Chase Variance

Displays source input accounts for one target input account, showing variances over two periods or categories.

Runs for

All Oracle Hyperion Financial Data Quality Management, Enterprise Edition locations

Parameters

Target Account, Category 1, Period 1, Category 2, Period 2.

Query

Account Chase Variance

Template

Account Chase Variance.rtf

Trial Balance Variance

Shows source input accounts, subtotaled by target accounts, showing variances over two periods or categories.

Runs for

Current Oracle Hyperion Financial Data Quality Management, Enterprise Edition location

Parameters

Category 1, Period 1, Category 2, Period 2

Query

Trial Balance Variance

Template

TB Variance.rtf

A

FDME E Web Service

This appendix describes the Oracle Hyperion Financial Data Quality Management, Enterprise Edition web service.

Note:

It is recommended that you use batch scripts rather than the web services, See [Working with Batch Scripts](#).

Access the FDMEE Web Services WSDL at <http://server:19000/oracle-epm-erpi-webservices/RuleService?wsdl>, where *Host Name* is the name of the host where Workspace is deployed.

To use the Web Service, configure the Oracle Web Services Manager. For more details, refer to *Configuring Oracle Web Services Manager for EPM System Products* in the *Oracle Enterprise Performance Management System Deployment Options Guide*.

executeDataRule

The executeDataRule method executes a data rule after validating the parameters passed with the rule. If the parameter is invalid, error messages are displayed.

Table A-1 executeDataRule web service

Parameter Name	Valid Values	Associated Error Codes	Comments
String pUserName		AIF_ERR_00002:Invalid User Name/Password combination.	For these three parameters pass the username and password or a valid sso_token.
String pPassword		AIF_ERR_00001:Invalid Token	
String pSSOToken			
String plImportFromSource	Y/N	AIF_ERR_00005:Invalid value for Import From Source Parameter	
String pExportToTarget	Y/N	AIF_ERR_00006:Invalid value for Export to Target Parameter	
String pRuleName		AIF_ERR_00004:Invalid Data Rule Name	The name of the data rule for a location.
String pExecutionMode	INCREMENTAL/ SNAPSHOT	AIF_ERR_00007:Invalid value for Execution mode	Data rule execution mode
String pLoadExchangeRateFlag	Y/N	AIF_ERR_00008:Invalid value for Load Exchange Rate Flag	This parameter can be used only in case of multi-currency applications.

Table A-1 (Cont.) executeDataRule web service

Parameter Name	Valid Values	Associated Error Codes	Comments
String pStartPeriodName	Start Period name	AIF_ERR_00010:Invalid Period Keys	This parameter represents where the execution should take place synchronously or asynchronously. Value of True executes the job synchronously.
String pEndPeriodName	End Period name	AIF_ERR_00010:Invalid Period Keys	
String pSyncMode	TRUE/FALSE	AIF_ERR_00009:Invalid value for sync mode	

executeMetadataRule

The executeMetadataRule method executes a metadata rule after validating the locationName parameter. If the parameter is invalid, error messages are displayed.

Table A-2 executeMetadataRule web service

Parameter Name	Valid Values	Associated Error Codes	Comments
String pUserName		AIF_COMMON_ERR_00001:Invalid token.	Of these three parameters pass either the username and password or a valid sso_token.
String pPassword		AIF_COMMON_ERR_00002:Invalid User Name/Password combination.	
String pSSOToken			
String locationName	TRUE/FALSE	AIF_ERR_00015: Invalid Location Name	The location name
String pSyncMode		AIF_ODI_ERR_00002:Invalid value passed for synchronous mode parameter.	This parameter represents where the execution should take place synchronously or asynchronously. Value of True executes the job synchronously.

getDataRuleNames

The getDataRuleNames method retrieves a list of data rule names for the location. If no location is passed, an empty array is returned.

Table A-3 etDataRuleNames web service

Parameter Name	Valid Values	Associated Error Codes	Comments
String pUserName String pPassword String pSSOToken		AIF_COMMON_ERR_00002:Invalid User Name/Password combination. AIF_COMMON_ERR_00001:Invalid token.	Fpr these three parameters pass either the username and password or a valid sso_token.
String locationName			The location name.

getLocationNames

Provides a list of locations for the executeMetaDataRule web service.

Table A-4 getLocationNames web service

Parameter Name	Valid Values	Associated Error Codes	Comments
String pUserName String pPassword String pSSOToken		AIF_COMMON_ERR_00002:Invalid User Name/Password combination. AIF_COMMON_ERR_00001:Invalid token.	

getPeriodNames

Provides a list of period names for the pStartPeriodName and pEndperiodName parameters of the executeDataRule service.

Table A-5 get PeriodNames web service

Parameter Name	Valid Values	Associated Error Codes	Comments
String pUserName String pPassword String pSSOToken		AIF_COMMON_ERR_00002:Invalid User Name/Password combination. AIF_COMMON_ERR_00001:Invalid token.	

lockPOV

Returns true if this POV is locked, otherwise returns false for the RuleService web services.

Table A-6 lockPOV

Parameter Name	Valid Values	Associated Error Codes	Comments
String pLocationName String pPeriodName String pCategoryName	Accepts Location Name, Period Name, Category Name as input parameters.		

unlockPOV

Returns true if this POV is unlocked, otherwise returns false for the RuleService web services.

Table A-7 unlock POV

Parameter Name	Valid Values	Associated Error Codes	Comments
String pLocationName String pPeriodName String pCategoryName	Accepts Location Name, Period Name, Category Name as input parameters.		

B

FDMEE REST APIs

Use the Oracle Hyperion Financial Data Quality Management, Enterprise Edition REST (Representational State Transfer) APIs to run data rules, run batches, import and export data mapping, and to execute reports. These APIs provide an alternative to using the selected components in the web-based user interface.

Completing administrative tasks using FDMEE REST APIs is an alternative to using the user interface requires considerable technical and functional expertise. Only technically competent EPM Cloud Administrators and consultants should perform FDMEE Administrator administrative tasks.

For more information on using FDMEE and other Oracle Enterprise Performance Management Cloud REST APIs, see the REST API for Oracle Enterprise Performance Management Cloud Guide.

URL Structure for FDMEE

URL Structure

Use the following URL structure to access the Oracle Hyperion Financial Data Quality Management, Enterprise Edition REST resources:

```
https://<SERVICE_NAME>-<TENANT_NAME>.<dcx>.oraclecloud.com/aif/rest/{api_version}/{path}
```

Where:

api_version—API version you are developing with. The current REST API version for FDMEE is V1.

path—Identifies the resource

Running Data Rules

Executes an Oracle Hyperion Financial Data Quality Management, Enterprise Edition data load rule based on the start period and end period, and import or export options that you specify.

Prerequisites

- Data Rules: Data load rules define how Integrations load data from a file. You must have predefined data load rules to load data.
- You must have the required privileges to execute a specific data rule.

REST Resource

POST /aif/rest/{api_version}/jobs

Request

Supported Media Types: application/json

Parameters

The following table summarizes the client request.

Table B-1 Parameters

Name	Description	Type	Required	Default
api_version	Version of the API you are working with, such as V1	Path	Yes	None
jobType	should be set to "DATARULE"		Yes	None
jobName	The name of a data load rule defined in FDMEE. You should enclose the rule name in quotation marks if it contains a space.		Yes	None
startPeriod	The first period for which data is to be loaded. This period name must be defined in FDMEE period mapping.		Yes	None
endPeriod	The last period for which data is to be loaded. This period name must be defined in FDMEE period mapping.		Yes	None
importMode	determines how the data is imported into FDMEE. Acceptable values are: • APPEND to add to the existing POV data in FDMEE • REPLACE to delete the POV data and replace it with the data from the file • RECALCULATE to skip importing the data, but re-process the data with updated Mappings and Logic Accounts. • NONE to skip data import into FDMEE staging table		Yes	None

Table B-1 (Cont.) Parameters

Name	Description	Type	Required	Default
exportMode	determines how the data is exported into FDMEE. Acceptable values for an Oracle Hyperion Planning application are: • STORE_DATA to merge the data in the FDMEE staging table with the existing Planning data • ADD_DATA to add the data in the FDMEE staging table to Planning • SUBTRACT_DATA to subtract the data in the FDMEE staging table from existing Planning data • REPLACE_DATA to clear the POV data and replace it with data in the FDMEE staging table. The data is cleared for Scenario, Version, Year, Period, and Entity • NONE to skip data export from FDMEE to Planning Acceptable values for Oracle Financial Consolidation and Close Cloud and Oracle Tax Reporting Cloud applications are: • MERGE—By default, all data load is processed in the Merge mode. If data already existed in the application, the system overwrites the existing data with the new data from the load file. If data does not exist, the new data will be created. • REPLACE—The system first clears any existing data in the application for those referenced in the data load file. Then the system performs the data load in Merge mode. • NONE—Skips the data export from FDMEE to Oracle Financial Consolidation and Close Cloud		Yes	None
fileName	An optional file name. If you do not specify a file name, this API imports the data contained in the file name specified in the load data rule. The data file must already reside in the INBOX prior to data rule execution.		Yes	None

Example URL

https://<SERVICE_NAME>-<TENANT_NAME>.<dcx>.oraclecloud.com/aif/rest/V1/jobs

Example of Request Body

```
{
  "jobType": "DATARULE",
  "jobName": "aso to bso dr",
  "startPeriod": "Dec-18",
  "endPeriod": "Dec-18",
}
```

```

"importMode": "REPLACE",
"exportMode": "NONE",
"fileName": ""
}

```

Response

Supported Media Types: application/json

Table B-2 Parameters

Name	Description
status	Status of the job: -1 = in progress; 0 = success; 1 = error; 2 = cancel pending; 3 = cancelled; 4 = invalid parameter
jobStatus	A text representation of the job status, with one of the following values "RUNNING", "SUCCESS", "FAILED"
jobId	The process ID generated in FDMEE for the job
logFileName	Log File containing entries for this execution.
outputFileName	Name of the output file generated, if any.
processType	Type of the process executed. Will contain "COMM_LOAD_BALANCES" for all Data Rule executions
executedBy	Login name of the user used to execute the rule.
details	Returns the exception stack trace in case of an application error

Example of Response Body

The following shows an example of the response body in JSON format.

```

{
  "jobStatus": "RUNNING"
  "jobId": 2019
  "logFileName": "\\outbox\\logs\\Account Reconciliation Manager_2019.log"
  "outputFileName": null
  "processType": "COMM_LOAD_BALANCES"
  "executedBy": "admin"
  "status": -1
  "links": [1]
    0: {
      "rel": "self"
      "href": "https://<SERVICE_NAME>-<TENANT_NAME>.<dcX>.oraclecloud.com/aif/rest/V1/jobs/2019"
      "action": "GET"
    }
  "details": null
}

```

Running Batch Rules

Executes a batch of jobs that have been defined in Oracle Hyperion Financial Data Quality Management, Enterprise Edition .

Prerequisites

- The batch must be defined in FDMEE before it can be executed using the EPM Automate Utility.
- You must have the required privileges to execute a specific batch..

REST Resource

POST /aif/rest/{api_version}/jobs

Request

Supported Media Types: application/json

The following table summarizes the client request.

Table B-3 Parameters

Name	Description	Type	Required	Default
api_version	Version of the API you are working with, such as V1	Path	Yes	None
jobType	should be set to "BATCH"		Yes	None
jobName	The name of a batch defined in FDMEE.		Yes	None

Example URL

`https://<SERVICE_NAME>-<TENANT_NAME>.<dcX>.oraclecloud.com/aif/rest/V1/jobs`

Example of Request Body

```
{
  "jobType": "BATCH",
  "jobName": "BatchDataLoad"
}
```

Response

The following table summarizes the response parameters.

Table B-4 Parameters

Name	Description
status	Status of the job: -1 = in progress; 0 = success; 1 = error; 2 = cancel pending; 3 = cancelled; 4 = invalid parameter
jobStatus	A text representation of the job status, with one of the following values "RUNNING", "SUCCESS", "FAILED"
jobId	The process Id generated in FDMEE for the job
logFileName	Log File containing entries for this execution.
outputFileName	Name of the output file generated, if any.
processType	Type of the process executed. Will contain "COMM_BATCH" for all Data Rule executions
executedBy	Login name of the user used to execute the rule.
details	Returns the exception stack trace in case of an application error

Supported Media Types: application/json

Example of Response Body

The following shows an example of the response body in JSON format.

```
{
  "jobStatus": "SUCCESS"
"jobId": 2016
"logFileName": "\\outbox\\logs\\BATCH1_7595.log"
"outputFileName": null
"processType": "COMM_BATCH"
"executedBy": "admin"
"status": -1
"links": [1]
  0: {
    "rel": "self"
    "href": "https://<SERVICE_NAME>-<TENANT_NAME>.<dcX>.oraclecloud.com/aif/rest/V1/jobs/2016"
    "action": "GET"
  }
"details": null
}
```

For sample code, see the code samples included in [Running Data Rules](#).

Import Data Mapping

Member mappings are used to derive the target members for each dimension based on source value. Member mappings are referenced during the data load, enabling Oracle Hyperion Financial Data Quality Management, Enterprise Edition to determine how to dimensionalize the data that is loaded to the target application. Member mappings define relationships between source members and target dimension members within a single dimension. You must create a member mapping for each target dimension.

You can import member mappings from a selected Excel, .CSV or .TXT file. You can also create new mappings in a text file and import them. Import member mappings support merge or replace modes, along with validate or no validate options for target members.

REST Resource

POST /aif/rest/{api_version}/jobs

Request

Supported Media Types: application/json

The following table summarizes the client request.

Table B-5 Parameters

Name	Description	Type	Required	Default
api_version	Version of the API you are working with, such as V1	Path	Yes	None
jobType	The job type, MAPPINGIMPORT	Path	Yes	None

Table B-5 (Cont.) Parameters

Name	Description	Type	Required	Default
jobName	The dimension name for a specific dimension to import, such as ACCOUNT, or ALL to import all dimensions	Path	Yes	None
fileName	The file and path from which to import mappings. The file format can be .CSV, .TXT, .XLS, or .XLSX. The file must be uploaded prior to importing, either to the inbox or to a sub-directory of the inbox. Include the inbox in the file path, for example, inbox/BESSAPPJan-06.csv	Path	Yes	None
importMode	The import mode: MERGE to add new rules or replace existing rules, or REPLACE to clear prior mapping rules before import	Path	No	MERGE
validationMode	Whether to use validation mode, true or false An entry of true validates the target members against the target application; false loads the mapping file without any validations. Note that the validation process is resource intensive and takes longer than the validation mode of false; the option selected by most customers is false	Path	No	false
locationName	The FDMEE location where the mapping rules should be loaded; mapping rules are specific to a location in FDMEE	Path	No	None

Example of Request Body

The following shows an example of the request body in JSON format.

```
{
  "jobType": "MAPPINGIMPORT",
  "jobName": "ACCOUNT",
  "fileName": "inbox/BESSAPPJan-06.csv",
  "importMode": "MERGE",
  "validationMode": "false",
  "locationName": "BESSAPP"
}
```

For sample code, see the code samples included in [Running Data Rules](#).

Response

The following table summarizes the response parameters.

Table B-6 Parameters

Name	Description
jobId	The process ID generated in FDMEE for the job, such as 1880
jobStatus	The job status, such as RUNNING
logFileName	Log file containing entries for this execution, such as outbox/logs/BESSAPP-DB_1880.log
outputFileName	Name of the output file generated, if any, or else null
processType	Type of process executed, IMPORT_MAPPING
executedBy	Login name of the user used to execute the rule, such as admin
details	Returns the exception stack trace in case of an application error, or null

Supported Media Types: application/json

Parameters

Example of Response Body

The following shows an example of the response body in JSON format.

```
{
  "links":
  [
    0]
  "status": "-1"
  "details": "null"
  "jobId": "1880"
  "jobStatus": "RUNNING",
  "logFileName": "outbox/logs/BESSAPP-DB_1880.log",
  "outputFileName": "null",
  "processType": "IMPORT_MAPPING",
  "executedBy": "admin"
}
```

For sample code, see the code samples included in [Running Data Rules](#).

Export Data Mapping

Member mappings are used to derive the target members for each dimension based on source value. Member mappings are referenced during the data load, enabling Oracle Hyperion Financial Data Quality Management, Enterprise Edition to determine how to dimensionalize the data that is loaded to the target application. Member mappings define relationships between source members and target dimension members within a single dimension. You must create a member mapping for each target dimension.

You can export member mappings to a selected file of format .csv, .txt, .xls, or .xlsx.

REST Resource

POST /aif/rest/{api_version}/jobs

Request

Supported Media Types: application/json

The following table summarizes the client request.

Table B-7 Parameters

Name	Description	Type	Required	Default
api_version	Version of the API you are working with, such as V1	Path	Yes	None
jobType	The job type, MAPPINGEXPORT	Path	Yes	None
jobName	The dimension name for a specific dimension to import, such as ACCOUNT, or ALL to import all dimensions	Path	Yes	None
fileName	The file and path from which to export mappings. The file format can be .CSV, .TXT, .XLS, or .XLSX. Include the outbox in the file path, for example, outbox/BESSAPPJan-06.csv	Path	Yes	None
locationName	The name of the location, such as BESSAPP	Path	Yes	None

Example of Request Body

The following shows an example of the request body in JSON format.

```
{
  "jobType": "MAPPINGIMPORT",
  "jobName": "ACCOUNT",
  "fileName": "outbox/BESSAPPJan-06.csv",
  "locationName": "BESSAPP"
}
```

For sample code, see the code samples included in [Running Data Rules](#).

Response

The following table summarizes the response parameters.

Table B-8 Parameters

Name	Description
jobId	The process ID generated in FDMEE for the job, such as 1881
jobStatus	The job status, such as SUCCESS
logFileName	Log file containing entries for this execution, such as outbox/logs/BESSAPP-DB_1881.log
outputFileName	Name of the output file generated, such as outbox/BESSAPPJan-06.csv
processType	The type of process executed, EXPORT_MAPPING
executedBy	Login name of the user used to execute the rule, such as admin
details	Returns the exception stack trace in case of an application error, or else null

Supported Media Types: application/json

Example of Response Body

The following shows an example of the response body in JSON format.

```
{
  "links":
  [
    0]
  "status": "0",
  "details": "null",
  "jobId": "1881",
  "jobStatus": "SUCCESS",
  "logFileName": "outbox/logs/BESSAPP-DB_1881.log",
  "outputFileName": "outbox/BESSAPPJan-06.csv",
  "processType": "EXPORT_MAPPING",
  "executedBy": "admin"
}
```

For sample code, see the code samples included in [Running Data Rules](#).

Execute Reports

The Oracle Hyperion Financial Data Quality Management, Enterprise Edition reporting framework represents a unified solution that incorporates source and target data, templates, and user-defined SQL queries. Templates, created in Oracle Business Intelligence Publisher, consume data in XML format and generate reports dynamically. You can add SQL queries to extract data from tables, or couple them with the report parameters to extend the definition of a standard report. FDMEE reports can be generated as PDF, Excel, Word, or HTML output.

REST Resource

POST /aif/rest/{api_version}/jobs

Request

Supported Media Types: application/json

The following table summarizes the client request.

Table B-9 Parameters

Name	Description	Type	Required	Default
api_version	Version of the API you are working with, such as V1	Path	Yes	None
jobType	The job type, REPORT	Path	Yes	None
jobName	The name of the report to be executed, such as Dimension Map For POV (Dimension, Cat, Per)	Path	Yes	None
reportFormatType	The file format of the report, pdf, xlsx, html, or excel	Path	Yes	pdf
parameters	Can vary in count and values based on the report	Path	Yes	None

Table B-9 (Cont.) Parameters

Name	Description	Type	Required	Default
Location	The location of the report, such as Comma_Vision	Path	Yes	None

Example of Request Body

The following shows an example of the request body in JSON format.

```
{
  "jobType": "REPORT",
  "jobName": "Dimension Map For POV (Dimension, Cat, Per)",
  "reportFormatType": "PDF",
  "parameters": {
    "Dimension Name": "ENTITY",
    "Category": "Actual",
    "Period": "Jan15",
    "Location": "Comma_Vision"
  }
}
```

For sample code, see the code samples included in [Running Data Rules](#).

Response

The following table summarizes the response parameters.

Table B-10 Parameters

Name	Description
jobId	The process ID generated in FDMEE for the job, such as 1885
status	The job status, such as RUNNING
logFileName	Log file containing entries for this execution, such as outbox\logs \BESSAPP-DB_1885.log
outputFileName	Name of the output file generated; you can use this name to download the report
processType	Type of process executed, EXECUTE_REPORT
executedBy	Login name of the user used to execute the rule, such as admin
details	Returns the exception stack trace in case of an application error, or null

Supported Media Types: application/json

Parameters**Example of Response Body**

The following shows an example of the response body in JSON format.

```
{
  "links":
  [
    0]
}
```

```
"status": "-1",  
"details": "null",  
"jobId": "1885",  
"jobStatus": "RUNNING",  
"logFileName": "outbox/logs/1885.log",  
"outputFileName": "outbox/reports",  
"processType": "EXECUTE_REPORT",  
"executedBy": "admin"  
}
```

For sample code, see the code samples included in [Running Data Rules](#).

C

Source System Tables Used by FDMEE

This section lists the source system tables used by Oracle Hyperion Financial Data Quality Management, Enterprise Edition. FDMEE reads all tables listed and writes to GL_BUDGET_INTERFACE and GL_TRACK_DELTA_BALANCES.

It also describes how to create synonyms for Oracle E-Business Suite tables.

Fusion and E-Business Suite Source System Tables

These Fusion and E-Business Suite source system tables are used by Oracle Hyperion Financial Data Quality Management, Enterprise Edition. All tables require read privileges, unless noted otherwise.

Table C-1 E-Business Suite Source System Tables

Table/View Name	Schema	Object Type	Privilege	Comments
FND_FLEX_VALIDATION_QUALIFIERS	APPLSYS	Table	Read only	
FND_FLEX_VALIDATION_TABLES	APPLSYS	Table	Read only	
FND_FLEX_VALUES	APPLSYS	View	Read only	
FND_FLEX_VALUES_TL	APPLSYS	View	Read only	
FND_FLEX_VALUE_CHILDREN_V	APPS	View	Read only	View based on FND_FLEX_VALUE_NORM_HIERARCHY, FND_FLEX_VALUE_SETS, and FND_FLEX_VALUES_VL
FND_FLEX_VALUE_NORM_HIERARCHY	APPLSYS	Table	Read only	
FND_FLEX_VALUE_SETS	APPLSYS	Table	Read only	
FND_FORM_FUNCTIONS	APPLSYS	Table	Read only	
FND_ID_FLEXES	APPLSYS	Table	Read only	
FND_ID_FLEX_SEGMENTS	APPLSYS	Table	Read only	
FND_ID_FLEX_SEGMENTS_TL	APPLSYS	Table	Read only	
FND_ID_FLEX_STRUCTURES	APPLSYS	Table	Read only	
FND_ID_FLEX_STRUCTURES_TL	APPLSYS	Table	Read only	

Table C-1 (Cont.) E-Business Suite Source System Tables

Table/View Name	Schema	Object Type	Privilege	Comments
FND_KF_LABELED_SEGMENTS	FUSION	Table	Read-only	
FND_KF_SEGMENT_LABELS_B	FUSION	Table	Read-only	
FND_LANGUAGES	APPLSYS	Table	Read only	
FND_MESSAGES_TL	FUSION	Table	Read-only	
FND_NEW_MESSAGES	APPLSYS	Table	Read only	
FND_NUMBER	APPLSYS	PL/SQL Package	Execute	
FND_RESPONSIBILITY	APPLSYS	Table	Read only	
FND_RESPONSIBILITY_TL	APPLSYS	Table	Read only	
FND_SEGMENT_ATRIBUTE_VALUES	APPLSYS	Table	Read only	
GL_BALANCES	GL	Table	Read only	
GL_BALANCES_DEBITA	GL	Table	Read only	
GL_BUDGETS	GL	Table	Read only	
GL_BUDGET_ENTITIES	GL	Table	Read only	
GL_BUDGET_INTERFACE	GL	Table	Read and write privileges are required.	
GL_BUDGET_VERSIONS	GL	Table	Read only	
GL_CODE_COMBINATIONS	GL	Table	Read only	
GL_CODE_COMBINATIONS_KFV	APPS	View	Read only	View based on GL_CODE_COMBINATIONS
GL_DAILY_BALANCES	GL	Table	Read only	
GL_DAILY_CONVERSION_TYPES	GL	Table	Read only	
GL_DAILY_CONVERSION_TYPES_V	APPS	View	Read only	View based on GL_DAILY_CONVERSION_TYPES
GL_DAILY_RATES	GL	Table	Read only	
GL_ENCUMBRANCE_TYPES	GL	Table	Read only	
GL_INTERFACE	GL	Table	Read/Write	
GL_JE_CATEGORIES	GL	Table	Read only	
GL_JE_CATEGORIES_TL	GL	Table	Read only	

Table C-1 (Cont.) E-Business Suite Source System Tables

Table/View Name	Schema	Object Type	Privilege	Comments
GL_JE_SOURCES_ TL	GL	Table	Read only	
GL_LEDGERS		Table	Read only	Table (R12 only)
GL_PERIODS	GL	Table	Read only	
GL_PERIOD_SETS	GL	Table	Read only	
GL_PERIOD_STATU SES	GL	Table	Read only	
GL_PERIOD_TYPES	GL	Table	Read only	
GL_PERIOD_TYPES _V	APPS	View	Read only	View based on GL_PERIOD_TYPES
GL_SETS_OF_BOO KS	GL/APPS	View	Read only	Table (11i)/View based on GL_LEDGERS (R12)
GL_STAT_ACCOUN T_UOM	GL	Table	Read only	
GL_TRACK_DELTA_ BALANCES	GL	Table	Read/Write	Read and write privileges are required.

PeopleSoft Enterprise Financial Management Source System Tables

These PeopleSoft Enterprise Financial Management source system tables are used by Oracle Hyperion Financial Data Quality Management, Enterprise Edition. All tables require read privileges, unless noted otherwise.

Table C-2 PeopleSoft Enterprise Financial Management Source System Tables

Table/View Name	Table/View Name	Table/View Name	Table/View Name
PS_AFFILIATE_LAN G	PS_CHARTFIELD1_T BL	PS_LEDGER	PSDBFLDLABL
PS_AFFILIATE_VW	PS_CHARTFIELD2_T BL	PS_LEDGER_BUDG	PSDBFLDLABLLANG
PS_AFFINTRA1_LAN G	PS_CHARTFIELD3_T BL	PS_LOCATION_TBL	PSKEYDEFN
PS_AFFINTRA1_VW	PS_CLASS_CF_LAN G	PS_NAMES	PSOPTIONS
PS_AFFINTRA2_LAN G	PS_CLASS_CF_TBL	PS_OPER_UNIT_LA NG	PSRECDEFN
PS_AFFINTRA2_VW	PS_DEPT_TBL	PS_OPER_UNIT_TBL	PSRECFIELD
PS_ALTACCT_LANG	PS_DEPT_TBL_LAN G	PS_PAYGROUP_TBL	PS_TREE_NODE_TB L
PS_ALTACCT_TBL	PS_EARNINGS_TBL	PS_PC_BU_GL_VW	PS_TREE_NODE_LA NG
PS_BD_SCENARIO_ TBL	PS_FS_CF_TEMPLA TE	PS_PC_INT_TMPL_G L	PSTREEDEFN

Table C-2 (Cont.) PeopleSoft Enterprise Financial Management Source System Tables

Table/View Name	Table/View Name	Table/View Name	Table/View Name
PS_BOOK_CODE_TBL	PS_FS_FLD_PROMPT	PS_POSITION_DATA	PSTREEDEFNLANG
PS_BU_BOOK_TBL	PS_FUND_LANG	PS_PROD_TBL_LANG	PSTREELEAF
PS_BU_LED_GRP_TBL	PS_FUND_TBL	PS_PRODUCT_TBL	PSTREENODE
PS_BUD_REF_LANG	PS_GL_ACCOUNT_LANG	PS_PROGRAM_LANG	PSTREESTRCT
PS_BUD_REF_TBL	PS_GL_ACCOUNT_TBL	PS_PROGRAM_TBL	PSXLATITEM
PS_BUL_CNTL_BUD	PS_HPYPB_ACCT_LN (Read and write privileges are required.)	PS_PROJECT	PSXLATITEMLANG
PS_BUS_UNIT_LANG	PS_HYP_KK_BD_HDR (Write privileges are required. Used for PeopleSoft Commitment Control)	PS_PROJECT_TBL	
PS_BUS_UNIT_TBL_FS	PS_HYP_KK_BD_LN (Write privileges are required. Used for PeopleSoft Commitment Control)	PS_REC_GROUP_REC	
PS_BUS_UNIT_TBL_GL	PS_JOB	PS_REC_GROUP_TBL	
	PS_JOBCODE_TBL	PS_RT_DFLT_VW	
PS_BUS_UNIT_TBL_HR		PS_RT_INDEX_TBL	
PS_CAL_ADJP_TBL	PS_LED_DEFN_LANG	PS_RT_TYPE_TBL	
PS_CAL_DEFN_TBL	PS_LED_DEFN_TBL	PS_SET_CNTRL_REC	
PS_CAL_DETP_TBL	PS_LED_GRP_LANG	PS_SET_CNTRL_TBL	
PS_CAL_SUMP_TBL	PS_LED_GRP_LED_TBL	PS_SET_CNTRL_TREE	
PS_CF1_LANG	PS_LED_GRP_TBL	PS_SETID_TBL	
PS_CF2_LANG	PS_LED_TMPLT_TBL		
PS_CF3_LANG			

PeopleSoft Human Capital Management Source System Tables

These PeopleSoft Human Capital Management source system tables are used by Oracle Hyperion Financial Data Quality Management, Enterprise Edition. All tables require read privileges.

Table C-3 PeopleSoft Human Capital Management Source System Tables

Table/View Name	Table/View Name	Table/View Name	Table/View Name
PS_ACCT_CD_TBL	PS_CHARTFIELD3_TBL	PS_JOB_EARNS_DIS_T	PS_SAL_PLAN_TBL
PS_AFFILIATE_VW	PS_CLASS_CF_TBL	PS_JOBCODE_TBL	PS_SAL_STEP_TBL
PS_ALTACCT_TBL	PS_DEPT_BUDGET_ERN	PS_LOCATION_TBL	PS_SET_CNTRL_REC
PS_BENEF_PLAN_TBL	PS_DEPT_CF_TBL	PS_NAMES	PS_UNION_TBL
PS_BP_JOB	PS_DEPT_TBL	PS_OPER_UNIT_TBL	PSOPTIONS
PS_BUD_REF_TBL	PS_EARNINGS_TBL	PS_PAYGROUP_TBL	PSRECFIELD
PS_BUS_UNIT_TBL_FS	PS_FREQUENCY_TBL	PS_POSITION_DATA	PSXLATITEM
PS_BUS_UNIT_TBL_GL	PS_FUND_TBL	PS_PRODUCT_TBL	PSXLATITEMLANG
PS_BUS_UNIT_TBL_HR	PS_GL_ACCOUNT_TBL	PS_PROGRAM_TBL	
PS_CHARTFIELD1_TBL	PS_GL_ACCOUNT_VW	PS_PROJECT_TBL	
PS_CHARTFIELD2_TBL	PS_JOB	PS_SAL_GRADE_TBL	

PeopleSoft Commitment Control Source System Tables

These PeopleSoft Commitment Control source system tables are used by Oracle Hyperion Financial Data Quality Management, Enterprise Edition. All tables require read privileges.

Table C-4 PeopleSoft Commitment Control Source System Tables

Table/View Name
PS_KK_BUDGET_TYPE
PS_KK_SUBTYPE
PS_KK_FILTER
PS_KK_KEY_CF
PS_KK_BD_OFFSET
PS_CAL_BP_TBL
PS_LEDGER_KK

Table C-4 (Cont.) PeopleSoft Commitment Control Source System Tables

Table/View Name
PS_HYP_KK_BD_HDR (Used for PeopleSoft Commitment Control)
PS_HYP_KK_BD_LN (Write privileges are required. Used for PeopleSoft Commitment Control)

Creating Synonyms for E-Business Suite Tables

If a user other than the "apps" user is referenced, you can create synonyms for Oracle E-Business Suite tables.



Note:

Synonyms are required for a PeopleSoft setup since PeopleSoft administrators typically do not provide PeopleSoft schema applications to users for configurations.

To create synonyms for E-Business Suite tables:

1. Create synonyms for the E-Business Suite tables listed in this appendix.
2. Assign read access to all the synonyms that you created.
3. Assign write access to the following tables:
 - GL_Interface
 - GL_Budget_Interface
 - GL_Track_Delta_Balances

In ODI Topology for EBS_Data_Server, you can use the synonym for both the user and the schema.

D

Creating an Alternate Schema in an Enterprise Resource Planning (ERP) Source System

Prebuilt integrations to an Enterprise Resource Planning (ERP) source system in Oracle Hyperion Financial Data Quality Management, Enterprise Edition use a basic filter for data extraction, and assume that appropriate security has been defined to enable access by FDMEE. In some environments, direct access to the source systems tables is prohibited because of system policies, or you want to define a source system filter that is more detailed than what is provided in the FDMEE user interface.

For example, in the Account Reconciliation Manager (ARM) you want to extract balance sheet accounts or active accounts only from the source system. To do this, create an alternate schema in the source system. This method provides a desired level of security, a different source system filter, or both.

To create an alternate hierarchy for Oracle E-Business Suite and Peoplesoft (PSFT) systems:

1. Create a new schema or user in the source system database.
2. Grant **SELECT** or **SELECT/INSERT** access to the list of source tables used by FDMEE to the new schema.

Refer to the source table list provided in [Source System Tables Used by FDMEE](#).

3. For E-Business Suite systems, create a new view named **GL_CODE_COMBINATIONS**, which includes the desired source filter.

For **Peoplesoft** systems, create a view using the **PS_LEDGER** table.

All columns from the source table must be included in the view.

4. Create synonyms for all remaining source tables that FDMEE references from the source system in the new schema.

Synonyms point to the base tables in the source system schema.

5. Update **Oracle Data Integrator (ODI)** to use the new schema in the physical schema for the related data server.

For example, the view created on the EBS GL_CODE_COMBINATIONS table may look like:

```
CREATE VIEW GL_COMBINATIONS (SEGMENT1, SEGMENT2,...,...)

SELECT SEGMENT1, SEGMENT2,...

FROM APPS.GL_CODE_COMBINATIONS

WHERE "ADD FILTERS"
```

Any views created in the new schema or synonyms must use the exact same name as specified in [Source System Tables Used by FDMEE](#). Because the table and view

names are the same as the core schema, FDMEE can access the updated contents with a change to the schema specification in ODI.

For SAP or JDE source systems, change the filter definition in the adapter rather than creating an alternate schema.

Before making any changes, it is recommended that you contact Oracle support to review the process to ensure a smooth update to the system.

E

Staging Tables

This section describes the Oracle Hyperion Financial Data Quality Management, Enterprise Edition staging tables.

FDMEEE Staging Tables

The following sections describe Oracle Hyperion Financial Data Quality Management, Enterprise Edition staging tables.

Note:

Oracle provides powerful tools you can use to insert, update, and delete information. But, if you use Oracle tools like SQL*Plus or Oracle Data Browser to modify Oracle Applications data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data. Because Oracle Applications tables are interrelated, any change you make using an Oracle Applications form can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications forms, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle Applications. When you use Oracle Applications forms to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. But, if you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.

Consequently, we **STRONGLY RECOMMEND** that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications tables, unless we tell you to do so in our installation, implementation, or open interface guides.

Staging Table Used for Import from Source

Table E-1 Staging Table Used for Import From Source

Table/View Name	Description
TDATASEG_T	Temporary data table used for transforming data from source to target in a data rule execution. To ensure read consistency, data is extracted from the corresponding staging table (for example, AIF_EBS_GL_BALANCES_STG, AIF_EBS_GL_DAILY_BAL_STG, or AIF_PS_LEDGER) and copied to TDATASEG_T (partitioned by LOADID), which is then used for transforming source data to target data based on the data mappings specified in TDATAMAP_T.

Data Tables Used for Drill Through

Table E-2 Data Table Used for Drill Through

Table/View Name	Description
TDATASEG	Audit table describing the data transformations in a data rule execution. Only in a successful data rule execution data transformations stored in TDATASEG_T be copied over to TDATASEG.

Views Used for Export to Target

Table E-3 Export to Target Views

Table/View Name	Description
AIF_HS_BALANCES	Interface view used for exporting data to a target application in a data rule execution. This view queries the TDATASEG_T table directly by summarizing the target amounts across the unique target dimension member combinations.
AIF_EBS_GL_INTERFACE_V	Interface view used for exporting data back to the Fusion and E-Business Suite general ledger system in a data load to write back rule execution. This view queries the TDATASEG table directly by summarizing the target amounts across the unique target segment value combinations.

Table E-3 (Cont.) Export to Target Views

Table/View Name	Description
AIF_PS_GL_INTERFACE_V	Interface view used for exporting data back to the PeopleSoft Enterprise Financial Management system in a data load rule to write back rule execution. This view queries the TDATASEG table directly by summarizing the target amounts across the unique target ChartField value combinations.

PeopleSoft Human Capital Management Staging Tables

The following sections describe the columns in each PeopleSoft Human Capital Management staging table used by Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

PS90HCM

Table E-4 AIF_PS_POSITION_DATA_STG Table

Column Name	Column Description
DESCR	Position Description

Table E-5 AIF_PS_JOBCODE_TBL_STG Table

Column Name	Column Description
COMP_FREQUENCY	Compensation Frequency
DESCR	Description
EFFDT	Effective Date
EFF_STATUS	Status
STD_HOURS	Standard Hours

Table E-6 AIF_PS_JOB_STG Table

Column Name	Column Description
ABSENCE_SYSTEM_CD	Absence System
ACCDNT_CD_FRA	Work Accident Code
ACCT_CD	Account Code
ADDS_TO_FTE_ACTUAL	Adds to FTE Actual Count
ANNL_BENEF_BASE_RT	Annual Benefit Base Rate
APPT_TYPE	Appointment Type
ASGN_END_DT	Assignment End Date
ASGN_START_DT	Assignment Start Date
AUTO_END_FLG	Auto End Job
BARG_UNIT	Bargaining Unit

Table E-6 (Cont.) AIF_PS_JOB_STG Table

Column Name	Column Description
BAS_ACTION	Benefits Administration Action
BAS_GROUP_ID	Benefits Administration Group ID
BENEFIT_SYSTEM	Benefits System
BORDER_WALKER	Cross Border Worker
CHANGE_AMT	Change Amount
CHANGE_PCT	Change Percent
CLASS_INDC	Classified/Unclassified Indicator
COBRA_ACTION	COBRA Action
COMPANY	Company
COMPRATE	Compensation Rate
COMP_FREQUENCY	Compensation Frequency
CONTRACT_NUM	Contract Number
CTG_RATE	Category Rate
CURRENCY_CD1	Currency Code 1
CUR_RT_TYPE	Currency Rate type
DAILY_RT	Daily Rate
DEPT_ENTRY_DT	Department Entry Date
DIRECTLY_TIPPED	Directly Tipped
EARNES_DIST_TYPE	Earnings Distribution Type
EEO_CLASS	EEO Class
ELIG_CONFIG1	Eligibility Configuration Field 1
ELIG_CONFIG2	Eligibility Configuration Field 2
ELIG_CONFIG3	Eligibility Configuration Field 3
ELIG_CONFIG4	Eligibility Configuration Field 4
ELIG_CONFIG5	Eligibility Configuration Field 5
ELIG_CONFIG6	Eligibility Configuration Field 6
ELIG_CONFIG7	Eligibility Configuration Field 7
ELIG_CONFIG8	Eligibility Configuration Field 8
ELIG_CONFIG9	Eligibility Configuration Field 9
EMPL_CTG	Employee Category
EMPL_CTG_L1	Employee Subcategory
EMPL_CTG_L2	Employee Subcategory 2
ENCUMB_OVERRIDE	Encumbrance Override
ENTRY_DATE	Entry Date
ESTABID	Establishment ID
EXEMPT_HOURS_MONTH	Exempted Hours Month
EXEMPT_JOB_LBR	Exempted
EXPECTED_END_DATE	Expected Job End Date
EXPECTED_RETURN_DT	Expected Return Date
FLSA_STATUS	FLSA Status
FORCE_PUBLISH	Force Publish
FUNCTION_CD	Function Code
GL_PAY_TYPE	GL Pay type

Table E-6 (Cont.) AIF_PS_JOB_STG Table

Column Name	Column Description
GP_ASOF_DT_EXG_RT	Use Rate As Of
GP_DFLT_CURRTTYP	Use Pay Group Rate Type
GP_DFLT_ELIG_GRP	Use Pay Group Eligibility
GP_DFLT_EXRTDT	Use Pay Group As Of Date
GP_ELIG_GRP	Eligibility Group
GP_PAYGROUP	Pay Group
GRADE_ENTRY_DT	Salary Grade Entry Date
HIRE_DT	Hire Date
HOLIDAY_SCHEDULE	Holiday Schedule
HOURLY_RT	Hourly Rate
HOURLY_RT_FRA	Hourly Rate France
HR_STATUS	HR Status
INTERCTR_WRKS_CNCL	Works Council Role
	Role that the worker has in the union group
JOB_DATA_SRC_CD	Job Source Code
JOB_ENTRY_DT	Job Entry Date
JOB_INDICATOR	Job Indicator
LABOR_AGREEMENT	Labor Agreement
LABOR_FACILITY_ID	Labor Facility ID
LABOR_TYPE_GER	Labor Type Germany
LASTUPDDTTM	Last Update Date/Time
LASTUPDOPRID	Last Update User ID
LAST_DATE_WORKED	Last Date Worked
LAST_HIRE_DT	Last Hire Date
LAYOFF_EXEMPT_FLAG	Layoff Exempt Flag
LAYOFF_EXEMPT_RSN	Layoff Exempt Reason
LBR_FAC_ENTRY_DT	Labor Facility Entry Date
LDW_OVR	Override Last Date Worked
LST_ASGN_START_DT	Last Assignment Start Date
LUMP_SUM_PAY	Lump Sum Pay
MAIN_APPT_NUM_JPN	Main Appointment Number
MATRICULA_NBR	Matricula Number
MONTHLY_RT	Monthly Rate
OFFICER_CD	Officer Code
PAID_FTE	Paid FTE
PAID_HOURS	Paid Hours
PAID_HRS_FREQUENCY	Paid Hours Frequency
PAYGROUP	Pay Group
PAY_SYSTEM_FLG	Pay System Flag
PAY_UNION_FEE	Pay Union Fee
PERFORM_GROUP_GER	Perform Group Germany
PER_ORG	Organizational Relationship
POI_TYPE	Person of Interest Type

Table E-6 (Cont.) AIF_PS_JOB_STG Table

Column Name	Column Description
POSITION_ENTRY_DT	Position Entry Date
POSITION_OVERRIDE	Position Override
POSN_CHANGE_RECORD	Position Change Record
PRORATE_CNT_AMT	Prorate Contract Change Amount
REG_REGION	Regulatory Region
REPORTS_TO	Reports To
SETID_DEPT	SetID Department
SETID_JOBCODE	SetID Job Code
SETID_LBR_AGRMNT	SetID Labor Agreement
SETID_LOCATION	SetID Location
SETID_SALARY	SetID Salary
SETID_SUPV_LVL	SetID Supervisor Level
SHIFT	Shift
SHIFT_FACTOR	Shift Factor
SHIFT_RT	Shift Rate
SOC_SEC_RISK_CODE	Social Security Risk Code
SPK_COMM_ID_GER	Spokesman Committee ID
STD_HRS_FREQUENCY	Standard Hours Frequency
STEP_ENTRY_DT	Step Entry Date
SUPERVISOR_ID	Supervisor ID
SUPV_LVL_ID	Supervisor Level ID
TARIFF_AREA_GER	Tariff Area Germany
TARIFF_GER	Tariff Germany
TAX_LOCATION_CD	Tax Location Code
TERMINATION_DT	Termination Date
UNION_FEE_AMOUNT	Union Fee Amount
UNION_FEE_END_DT	Union Fee End Date
UNION_FEE_START_DT	Union Fee Start Date
UNION_FULL_PART	Union Participation
UNION_POS	Union Position
UNION_SENIORITY_DT	Union Seniority Date
VALUE_1_FRA	Value 1
VALUE_2_FRA	Value 2
VALUE_3_FRA	Value 3
VALUE_4_FRA	Value 4
VALUE_5_FRA	Value 5
WORK_DAY_HOURS	Work Day Hours
WPP_STOP_FLAG	Stop Wage Progression
WRKS_CNCL_FUNCTION	Work Council Function
WRKS_CNCL_ROLE_CHE	Work Council Role

Table E-7 AIF_PS_POSITION_DATA_STG Table

Column Name	Column Description
ADDS_TO_FTE_ACTUAL	Adds to FTE Actual Count
BARG_UNIT	Bargaining Unit
CLASS_INDC	Classified/Unclassified Indicator
COMPANY	Company
CONFIDENTIAL_POSN	Confidential Position
COUNTRY_CODE	Country Code
DESCRSHORT	Position Short Description
EG_ACADEMIC_RANK	Academic Rank
EG_GROUP	Service Calculation Group
ENCUMBER_INDC	Encumbrance Indicator
ENCUMB_SAL_AMT	Encumbrance Salary Amount
ENCUMB_SAL_OPTN	Encumbrance Salary Option
FLSA_STATUS	FLSA Status
FRI_HRS	Friday Hours
GRADE_FROM	Grade From
GRADE_TO	Grade To
GVT_AUDITED_BY_ID	Position Audited By
GVT_COMP_AREA	Competitive Area
GVT_COMP_LEVEL	Competitive Level
GVT_DRUG_TEST_REQ	Drug Test (Applicable)
GVT_DT_POSN_ESTAB	Date Position Established
GVT_EXEC_FIN_DISCL	Executive Financial Disclosure
GVT_FUND_SOURCE	Fund Source
GVT_INTEL_POSN	Intelligence Position
GVT_LAST_AUDIT_DT	Position Audit Date
GVT_LEO_POSITION	LEO/Fire Position
GVT_MAINT_REVIEWCD	Maintenance Review
GVT_MANTREV_RESULT	Maintenance Review Results
GVT_MOB_POSN_IND	Mobility Position
GVT_NFC_FUNCNTN_CD	NFC Function Code (NFC - National Finance Center)
GVT_NFC_PI_IND_OVR	Override NFC Indicator
GVT_NTE_DATE	Not to Exceed Date
GVT_OBLIGATED_TO	Obligated To ID
GVT_OBLIG_EXPIR_DT	Obligation Expiration
GVT_OCC_SERIES	Occupational Series
GVT_ORG_NUMCD1	Numeric Code 1
GVT_ORG_NUMCD2	Numeric Code 2
GVT_ORG_NUMCD3	Numeric Code 3
GVT_ORG_NUMCD4	Numeric Code 4
GVT_ORG_NUMCD5	Numeric Code 5
GVT_ORG_NUMCD6	Numeric Code 6
GVT_ORG_NUMCD7	Numeric Code 7
GVT_ORG_SUBAGENCY	Organizational Structure

Table E-7 (Cont.) AIF_PS_POSITION_DATA_STG Table

Column Name	Column Description
GVT_ORG_TTL_CD	Organization Position Title Code
GVT_ORG_TTL_DESCR	Organization Position Title
GVT_PAREN_TITLE	Parenthetical Title
GVT_PAR_NBR	Personnel Action Request Number
GVT_PAY_PLAN	Pay Plan
GVT_PERF_PLAN	Performance Plan
GVT_PI_UPD_IND	PI Update Indicator
GVT_POI	Personnel Office ID
GVT_POSN_LOC	Position Location
GVT_POSN_OCCUPIED	Position Occupied
GVT_POSN_SENS_CD	Sensitivity Code
GVT_PRESIDENT_APPT	Presidential Appointment Position
GVT_PROC_INTEG_PSN	Procurement Integration Position
GVT_STAFF_LINE	Staff/Line Position
GVT_STATUS_TYPE	Status Type
GVT_SUB_AGENCY	Sub-Agency
GVT_TARGET_GRADE	Target Grade
GVT_VACY_REVW_CD	Vacancy Review Code
GVT_WIP_STATUS	Work-In-Progress Status
GVT_WORK_SCHED	Work Schedule
HEALTH_CERTIFICATE	Health Certificate
INCLUDE_SALPLN_FLG	Include Salary Plan Flag
JOB_SHARE	Job Share
KEY_POSITION	Key Position
LANGUAGE_SKILL	Language Skill
LASTUPDDTTM	Last Update Date/Time
LASTUPDOPRID	Last Update User ID
LOCATION	Location
MAIL_DROP	Mail Drop
MANAGER_LEVEL	Manager Level
MILITARY_SERVICE	Military Service
MIL_RANK	Military Rank
MIL_SVC_COMP_CD	Military Service Component
MON_HRS	Monday Hours
ORGCODE	Organization Hierarchy Code
ORGCODE_FLAG	Organization Hierarchy Flag
PHONE	Phone
PKG_RULE_ID	Package Rule ID
PKG_TEMPLATE_ID	Package Template ID
POSITION_POOL_ID	Position Pool ID
REG_REGION	Regulatory Region
REPORTS_TO	Reports To
REPORT_DOTTED_LINE	Report Dotted Line

Table E-7 (Cont.) AIF_PS_POSITION_DATA_STG Table

Column Name	Column Description
SAT_HRS	Saturday Hours
SEASONAL	Seasonal
SEC_CLEARANCE_TYPE	Security Clearance Type
SHIFT	Shift
SIGN_AUTHORITY	Signing Authority
STATUS_DT	Status Date
STD_HRS_FREQUENCY	Standard Hours Frequency
STEP_FROM	Step From
STEP_TO	Step To
SUN_HRS	Sunday Hours
SUPV_LVL_ID	Supervisory Level ID
THURS_HRS	Thursday Hours
TRN_PROGRAM	Training Program
TUES_HRS	Tuesday Hours
UPDATE_INCUMBENTS	Update Incumbents
WED_HRS	Wednesday Hours

Table E-8 AIF_PS_SAL_GRADE_TBL_STG Table

Column Name	Column Description
DESCR	Grade Description
DESCRSHORT	Grade Description Short
EFFDT	Effective Date
EFF_STATUS	Effective Status
ELIGIBILITY_POINTS	Eligibility Points
EMPL_CTG	Employee Category
EMPL_CTG_L1	Employee Subcategory
EMPL_CTG_L2	Employee Subcategory 2
GRADE	Grade
GRADE_TITLE_JPN	Grade Title Japan
GVT_MAX_RT_BIWK	Maximum Pay Rate - Bi-weekly
GVT_MID_RT_BIWK	Midpoint Pay Rate - Bi-weekly
GVT_MIN_RT_BIWK	Minimum Pay Rate - Bi-weekly
LABOR_AGREEMENT	Labor Agreement
MAX_YRS_GRADE	Max Years Grade
MIN_GRD_AGE	Minimum Grade Age
MIN_YRS_GRADE	Minimum Years Grade
REG_REGION	Regulatory Region
REVIEW_COUNT	Review Count
SALARY_MATRIX_CD	Salary Matrix Code
SAL_ADMIN_PLAN	Salary Administration Plan
SETID	SetID
STEP_INCREM_ACCUM	Step Increment Accumulated

Table E-8 (Cont.) AIF_PS_SAL_GRADE_TBL_STG Table

Column Name	Column Description
STEP_INCREM_TYPE	Step Increment Type

Table E-9 AIF_PS_SAL_STEP_TBL_STG Table

Column Name	Column Description
APS_MAX_CLS_AUS	Maximum Classification
APS_MIN_CLS_AUS	Minimum Classification
CLASSN_CURRENT_AUS	Job Classification
DAILY_RT	Daily Rate
EFFDT	Effective Date
GRADE	Grade
GVT_BIWEEKLY_RT	Bi-weekly Rate
GVT_INCREM_DAYS	Days to Next Step Increment
GVT_INCREM_WEEKS	Weeks to Next Step Increment
GVT_STEP	US Federal Step
GVT_WGI_ADJ	Within Grade Increase (WGI) Adjustment
INCREM_HOURS	Increment Hours
INCREM_MONTHS	Increment Months
SAL_ADMIN_PLAN	Salary Administration Plan
SETID	SetID
STEP_DESCR	Step Description

Table E-10 AIF_PS_BP_JOB_STG Table

Column Name	Column Description
ACCDNT_CD_FRA	Account Code France
ACCT_CD	Account Code
ACTION	Action
ACTION_DT	Action Date
ACTION_REASON	Action Reason
ADDS_TO_FTE_ACTUAL	Adds to FTE Actual Count
ANNL_BENEF_BASE_RT	Annual Benefit Base Rate
ANNL_BEN_BASE_OVRD	Annual Benefit Base Overhead
ANNUAL_RT	Annual Rate
APPT_TYPE	Appointment Type
BARG_UNIT	Bargaining Unit
BAS_ACTION	Benefits Administration Action
BAS_GROUP_ID	Benefits Administration Group ID
BENEFIT_PROGRAM	Benefit Program
BENEFIT_SYSTEM	Benefit System
BEN_STATUS	Benefit Status
BORDER_WALKER	Cross Border Worker
BUSINESS_UNIT	Business Unit

Table E-10 (Cont.) AIF_PS_BP_JOB_STG Table

Column Name	Column Description
CHANGE_AMT	Change Amount
CHANGE_PCT	Change Percent
CLASS_INDC	Classified/Unclassified Indicator
COBRA_ACTION	COBRA Action
COMPANY	Company
COMPRATE	Compensation Rate
COMP_FREQUENCY	Compensation Frequency
CONTRACT_NUM	Contract Number
CTG_RATE	Category Rate
CURRENCY_CD	Currency Code
CURRENCY_CD1	Currency Code 1
CUR_RT_TYPE	Currency Rate type
DAILY_RT	Daily Rate
DEPTID	Department
DEPT_ENTRY_DT	Department Entry Date
DIRECTLY_TIPPED	Directly Tipped
EARN_DIST_TYPE	Earnings Distribution Type
EEO_CLASS	EEO_Class
EFFDT	Effective Date
EFFSEQ	Effective Sequence
ELIG_CONFIG1	Eligibility Configuration Field 1
ELIG_CONFIG2	Eligibility Configuration Field 2
ELIG_CONFIG3	Eligibility Configuration Field 3
ELIG_CONFIG4	Eligibility Configuration Field 4
ELIG_CONFIG5	Eligibility Configuration Field 5
ELIG_CONFIG6	Eligibility Configuration Field 6
ELIG_CONFIG7	Eligibility Configuration Field 7
ELIG_CONFIG8	Eligibility Configuration Field 8
ELIG_CONFIG9	Eligibility Configuration Field 9
EMPLID	Employee ID
EMPL_CLASS	Employee Classification
EMPL_CTG	Employee Category
EMPL_CTG_L1	Employee Subcategory
EMPL_CTG_L2	Employee Subcategory 2
EMPL_RCD	Employee Record Number
EMPL_STATUS	Employee Status
EMPL_TYPE	Employee Type
ENCUMB_OVERRIDE	Encumbrance Override
ENTRY_DATE	Entry Date
ESTABID	Establishment ID
EXEMPT_HOURS_MONTH	Exempted Hours Month
EXEMPT_JOB_LBR	Exempted
FICA_STATUS_EE	FICA Status

Table E-10 (Cont.) AIF_PS_BP_JOB_STG Table

Column Name	Column Description
FLSA_STATUS	FLSA Status
FORCE_PUBLISH_FTE	Force Publish FTE
FULL_PART_TIME	Full or Part-Time
FUNCTION_CD	Function Code
GL_PAY_TYPE	GL Pay type
GP_ASOF_DT_EXG_RT	Use Rate As Of
GP_DFLT_CURRTTYP	Use Pay Group Rate Type
GP_DFLT_ELIG_GRP	Use Pay Group Eligibility
GP_DFLT_EXRTDT	Use Pay Group As Of Date
GP_ELIG_GRP	Eligibility Group
GP_PAYGROUP	Pay Group
GRADE	Grade
GRADE_ENTRY_DT	Grade Entry Date
GVT_ANNL_RT_NO_LOC	Annuity Rate
GVT_ANNUITY_OFFSET	Annuity Offset
GVT_ANNUIT_COM_DT	Annuity Date
GVT_ANN_IND	Annuity
GVT_BASIC_LIFE_RED	Basic Life Red
GVT_BIWEEKLY_RT	Biweekly Rate
GVT_BW_RT_NO_LOC	Biweekly Rate No Location
GVT_CHANGE_FLAG	Change Flag
GVT_COMPRATE	Compensate Rate
GVT_CONT_EMPLID	Contract Employee ID
GVT_CSRS_FROZN_SVC	CSRS Frozen Service
GVT_DAILY_RT	Daily Rate
GVT_DED_PRORT_DT	DED Prorate Date
GVT_DETL_POSN_NBR	Position Number
GVT_DLY_RT_NO_LOC	Daily Rate No Location
GVT_EFFDT	Effective Date
GVT_EFFDT_PROPOSED	Effective Date Proposed
GVT_ELIG_FEHB	Eligibility FEHB
GVT_FEGLI	FEGLI Code
GVT_FEGLI_BASC_PCT	FEGLI Basic Percent
GVT_FEGLI_LIVING	FEGLI Living Benefits
GVT_FEGLI_OPT_PCT	FEGLI Opted Percent
GVT_FEHB_DT	FEHB Date
GVT_FEHB_PCT	FEHB Percent
GVT_FERS_COVERAGE	FERS Coverage
GVT_HRLY_RT_NO_LOC	Hourly Rate No Location
GVT_LEG_AUTH_1	Legal Authority 1
GVT_LEG_AUTH_2	Legal Authority 2
GVT_LEO_POSITION	LEO/Fire Position (LEO - Law enforcement officer)
GVT_LIVING_AMT	Living Amount

Table E-10 (Cont.) AIF_PS_BP_JOB_STG Table

Column Name	Column Description
GVT_LOCALITY_ADJ	Locality Adjustment
GVT_MNLY_RT_NO_LOC	Monthly Rate No Location
GVT_NID_CHANGE	NID Change
GVT_NOA_CODE	NOA Code
GVT_OTH_PAY_CHG	Other Pay Change
GVT_PAR_AUTH_D1	Authority 1 Description - Part 1
GVT_PAR_AUTH_D1_2	Authority 1 Description - Part 2
GVT_PAR_AUTH_D2	Authority 2 Description - Part 1
GVT_PAR_AUTH_D2_2	Authority 2 Description - Part 1
GVT_PAR_NTE_DATE	NTE Date
GVT_PAY_BASIS	Pay Basis
GVT_PAY_FLAG	Pay Flag
GVT_PAY_PLAN	Pay Plan
GVT_PAY_RATE_DETER	Pay Rate Determinant
GVT_PI_UPD_IND	PI Update Indicator
GVT_POI	Personnel Office ID
GVT_POSN_OCCUPIED	Position Occupied
GVT_PREV_RET_COVRG	Previous Retirement Coverage
GVT_RETIRE_PLAN	Retire Plan
GVT_RETRO_BSE_FLAG	Retrospective Base Flag
GVT_RETRO_DED_FLAG	Retrospective Deduction Flag
GVT_RETRO_FLAG	Retrospective Flag
GVT_RETRO_JOB_FLAG	Retrospective Job Flag
GVT_ROUTE_NEXT	Route to Next
GVT_RTND_GRADE	Retained Grade
GVT_RTND_GVT_STEP	Retained Government Step
GVT_RTND_PAY_PLAN	Retained Pay Plan
GVT_RTND_SAL_PLAN	Retained Salary Plan
GVT_RTND_STEP	Retained Step
GVT_S113G_CEILING	S113G Ceiling
GVT_SF52_NBR	SF52 Number
GVT_STATUS_TYPE	Status Type
GVT_STEP	Step
GVT_SUB_AGENCY	Sub Agency
GVT_TRANS_NBR	Transaction Number
GVT_TRANS_NBR_SEQ	Transaction Number Sequence
GVT_TSP_UPD_IND	TSP Status
GVT_TYPE_OF_APPT	Type of Appointment
GVT_WIP_STATUS	WIP Status
GVT_WORK_SCHED	Work Schedule
GVT_XFER_FROM_AGCY	Transfer From Agency
GVT_XFER_TO_AGCY	Transfer To Agency
HOLIDAY_SCHEDULE	Holiday Schedule

Table E-10 (Cont.) AIF_PS_BP_JOB_STG Table

Column Name	Column Description
HOURLY_RT	Hourly Rate
HOURLY_RT_FRA	Hourly Rate France
INTERCTR_WRKS_CNCL	Works Council Role
JOB_CODE	Job Code
JOB_DATA_SRC_CD	Job Source Code
JOB_ENTRY_DT	Job Entry Date
JOB_INDICATOR	Job Indicator
LABOR_AGREEMENT	Labor Agreement
LABOR_FACILITY_ID	Labor Facility ID
LABOR_TYPE_GER	Labor Type Germany
LAYOFF_EXEMPT_FLAG	Layoff Exempt Flag
LAYOFF_EXEMPT_RSN	Layoff Exempt Reason
LBR_FAC_ENTRY_DT	Labor Facility Entry Date
LOCATION	Location
LUMP_SUM_PAY	Lump Sum Pay
MAIN_APPT_NUM_JPN	Main Appointment Number
MATRICULA_NBR	Matricula Number
MONTHLY_RT	Monthly Rate
OFFICER_CD	Officer Code
PAID_FTE	Paid FTE
PAID_HOURS	Paid Hours
PAID_HRS_FREQUENCY	Paid Hours Frequency
PAYGROUP	Pay Group
PAY_SYSTEM_FLG	Pay System Flag
PAY_UNION_FEE	Pay Union Fee
PERFORM_GROUP_GER	Perform Group Germany
POSITION_ENTRY_DT	Position Entry Date
POSITION_NBR	Position Number
POSITION_OVERRIDE	Position Override
POSN_CHANGE_RECORD	Position Change Record
PRORATE_CNT_AMT	Prorate Contract Change Amount
REG_REGION	Regulatory Region
REG_TEMP	Regular or Temporary
REPORTS_TO	Reports To
SAL_ADMIN_PLAN	Salary Administration Plan
SETID_DEPT	SetID Department
SETID_JOB_CODE	SetID Job Code
SETID_LBR_AGRMNT	SetID Labor Agreement
SETID_LOCATION	SetID Location
SETID_SALARY	SetID Salary
SHIFT	Shift
SHIFT_FACTOR	Shift Factor
SHIFT_RT	Shift Rate

Table E-10 (Cont.) AIF_PS_BP_JOB_STG Table

Column Name	Column Description
SOC_SEC_RISK_CODE	Social Security Risk Code
SPK_COMM_ID_GER	Spokesman Committee ID
STD_HOURS	Standard Hours
STD_HRS_FREQUENCY	Standard Hours
STEP	Frequency Step
STEP_ENTRY_DT	Step Entry Date
SUPERVISOR_ID	Supervisor ID
TARIFF_AREA_GER	Tariff Area Germany
TARIFF_GER	Tariff Germany
TAX_LOCATION_CD	Tax Location Code
UNION_CD	Union Code
UNION_FEE_AMOUNT	Union Fee Amount
UNION_FEE_END_DT	Union Fee End Date
UNION_FEE_START_DT	Union Fee Start Date
UNION_FULL_PART	Union Participation
UNION_POS	Union Position
UNION_SENIORITY_DT	Union Seniority Date
UPDATE_PAYROLL	Update Payroll
VALUE_1_FRA	Value 1
VALUE_2_FRA	Value 2
VALUE_3_FRA	Value 3
VALUE_4_FRA	Value 4
VALUE_5_FRA	Value 5
WORK_DAY_HOURS	Work Day Hours
WPP_STOP_FLAG	Stop Wage Progression
WRKS_CNCL_FUNCTION	Work Council Function
WRKS_CNCL_ROLE_CHE	Work Council Role

CHARTFIELD

Table E-11 Table Type—CHARTFIELD

Table Name	Column Name	Column Description
AIF_PS_GL_ACCOUNT_TBL_STG	ACCOUNT	Account
AIF_PS_AFFILIATE_VW_STG	AFFILIATE	Affiliate
AIF_PS_ALTACCT_TBL_STG	ALTACCT	Alternate Account
AIF_PS_BUD_REF_TBL_STG	BUDGET_REF	Budget Ref
AIF_PS_CHARTFIELD1_TBL_STG	CHARTFIELD1	Chartfield 1
AIF_PS_CHARTFIELD2_TBL_STG	CHARTFIELD2	Chartfield 2

Table E-11 (Cont.) Table Type—CHARTFIELD

Table Name	Column Name	Column Description
AIF_PS_CHARTFIELD3_TBL_STG	CHARTFIELD3	Chartfield 3
AIF_PS_CLASS_CF_TBL_STG	CLASS_FLD	Class
AIF_PS_DEPT_CF_TBL_STG	DEPTID_CF	Department
AIF_PS_FUND_TBL_STG	FUND_CODE	Fund Code
AIF_PS_OPER_UNIT_TBL_STG	OPERATING_UNIT	Operating Unit
AIF_PS_PRODUCT_TBL_STG	PRODUCT	Product
AIF_PS_PROGRAM_TBL_STG	PROGRAM_CODE	Program
AIF_PS_PROJECT_TBL_STG	PROJECT	Project

F

Archiving the TDATEMAPSEG Table

When importing data, the mapping used to transform data is archived in the TDATEMAPSEG table for each point of view (POV). If you use a large number of maps, this table grows in size over a period of time and can create a performance issue during the data import process. In order to improve performance, you can archive the data from the TDATEMAPSEG table.

In release 11.1.1.2.4.220, a new view TDATEMAPSEG_ALL is available. This view is used by the user interface to display the mappings.

You can modify the view definition to include the TDATEMAPSEG and ARCHIVE tables. Below is a sample script set that shows how to archive the data and create the view in the Oracle database.

```
/* Archive the existing table */

RENAME TDATEMAPSEG TO TDATEMAPSEG_ARCHIVE1

;

/* Create a new table */

CREATE TABLE TDATEMAPSEG

AS

SELECT *

FROM TDATEMAPSEG_ARCHIVE1

WHERE 1=2

;

/* Drop existing view */

DROP VIEW TDATEMAPSEG_ALL

;

/* Create a view which is UNION of base table and archive table */

CREATE VIEW TDATEMAPSEG_ALL AS

SELECT * FROM TDATEMAPSEG

UNION ALL

SELECT * FROM TDATEMAPSEG_ARCHIVE1

;
```

A similar process can be used for the SQL Server. Multiple archive tables can be created over a period of time and the view can be modified as needed.

G

Working with Open Interface Adapters

An open interface adapter is used as a generic staging table in Oracle Hyperion Financial Data Quality Management, Enterprise Edition from which you can move or extract data using any system.

Setting up the Open Interface Adapter

The Open Interface Adapter is used to map data from the open interface table (AIF_OPEN_INTERFACE) to the Oracle Hyperion Financial Data Quality Management, Enterprise Edition staging tables. Typically, the Open Interface Adapter is used to load from a source not supported with a pre-packaged adapter. The Open Interface Adapter eliminates the need to process and manage data files, but the user is responsible for loading the table. Load methods include:

- Event script in FDMEE
- ODI
- SQL Loader
- anything

You configure this adapter using FDMEE to populate the staging table with the related data from the open interface table. You load the open interface table with the tool of your choice like a pl/sql routine, a sql loader script or a custom program. When you use ODI to load the open interface table, you can customize the Open Interface Adapter ODI package shipped with FDMEE to call a custom ODI interface to load the open interface table as a step in the data load process.

Using the Open Interface Table

The open interface table is called AIF_OPEN_INTERFACE. The table definition is as follows:

Table G-1 AIF_OPEN_INTERFACE Table Definitions

Column Name	Data Type	Comment
BATCH_NAME	VARCHAR2(200 BYTE)	Required. Used to filter the data
YEAR	NUMBER(15,0)	Optional. Required if period mapping is Explicit
PERIOD	VARCHAR2(30 BYTE)	Optional. Required if period mapping is Explicit
PERIOD_NUM	NUMBER(15,0)	Optional. Required if period mapping is Explicit
CURRENCY	VARCHAR2(10 BYTE)	Required.

Table G-1 (Cont.) AIF_OPEN_INTERFACE Table Definitions

Column Name	Data Type	Comment
DATAVIEW	VARCHAR2(8 BYTE)	Used for HFM applications. Assign value of YTD or Periodic. Default value is YTD
DESC1	VARCHAR2(75 BYTE)	Optional. Mapped to DESC1
DESC2	VARCHAR2(75 BYTE)	Optional. Mapped to DESC2
AMOUNT	NUMBER(29,12)	Required. Mapped to AMOUNT
AMOUNT_YTD	NUMBER(29,12)	Optional. Mapped to AMOUNT_YTD
AMOUNT_PTD	NUMBER(29,12)	Optional. Mapped to AMOUNT_PTD
COL01	VARCHAR2(75 BYTE)	Input Source Column 1
COL02	VARCHAR2(75 BYTE)	Input Source Column 2
COL03	VARCHAR2(75 BYTE)	Input Source Column 3
....		
COL30	VARCHAR2(75 BYTE)	Input Source Column 30
ATTR1	VARCHAR2(20 BYTE)	Optional. Mapped to ATTR1
ATTR2	VARCHAR2(20 BYTE)	Optional. Mapped to ATTR2
...		
ATTR13	VARCHAR2(20 BYTE)	Optional Mapped to ATTR13
ATTR14	VARCHAR2(20 BYTE)	Reserved. Do not map.

Loading Source Data using Custom Loaders

To load data using the Open Interface Table and Adapter:

1. Populate the Open Interface table as needed.
2. Create any new period mappings as needed.
3. Create a new source system:
 - a. On the **Setup** tab, under **Register**, then **Source System**.
 - b. Select **Add** to add a new source system.
 - c. Specify the **Name**, **Source System Type**, and **ODI Context Code**.
 The ODI context code refers to the context defined in Oracle Data Integrator. A context groups the source and target connection information.
 For the Source system Type, select "Others," and for the ODI Context Code, enter the default "GLOBAL." If the ODI has been set up differently from the defaults, enter the appropriate ODI Context Code if it is not "GLOBAL."
4. From **Source Adapter**, copy the Open Interface Adapter to a new adapter.
 - a. Provide a new adapter key.
 The adapter key is ten characters long. Once the adapter key has been created, it cannot be modified.
 - b. Change the "Adapter Name" so that it is different from the copied adapter name.

The adapter name is used in pick lists, and needs to be changed so that the copied adapter can be distinguished from the original adapter.

5. On the Import Format screen, define a new import format.
 - a. Select the source system created in Step 3 and the adapter from Step 4.
 - b. Map the source columns to the appropriate target dimensions.
6. When the import format mappings are complete, select **Regenerate ODI Scenario**.

This step generates the ODI Scenario based on the import format mappings.

7. On the **Locations** screen, define a new location using the import format.
8. On the **Data Rule** screen, define the new data rule.
 - a. Select the appropriate period mapping type.

Provide a value for the batch name. The batch name is used by the system as a filter on the open interface table. This enables multiple users to load data into a single interface table but then segregate the data by the batch name.

- b. Enter **Y** or **N** for the Delete Data After Import option.
9. Execute the data rule using one of three methods:
 - Data Rule Execution
 - Workbench
 - Batch Execution Script

Customizing ODI to Load Data from a Custom Source

You can modify the Open Interface Adapter to meet your business needs, and then once fully tested, move the updated content to a different environment for further testing or use in production. The steps to perform the customization and migration are listed below.

Use ODI Studio to perform the following steps. Before performing these steps, backup you work repository, custom project, and model.

When exporting and importing objects, use the following options:

- When exporting, select the **Child Components Export** option.
- When importing, use **Synonym Mode INSERT_UPDATE** as the Import Type.

To modify the Open Interface Adapter:

1. In the Oracle Hyperion Financial Data Quality Management, Enterprise Edition repository, export the **Open Interface Adapter Model** and **ERPI Target Model** folders.
2. In the FDMEE repository, export the **Open Interface Adapter Project**.
3. Create a new work repository with a new ID.

Use a repository ID of 900 or higher to avoid conflicts with Oracle shipped components. This is your development repository when performing customization. It is referred to as the "custom repository" in the following steps.

4. In the custom repository, import the model folders named: **Open Interface Adapter Model** and **ERPI Target Model**.

5. In the custom repository, import the **Open Interface Adapter Project**.
6. Perform any customizations in the custom repository.

When customizing objects, do not modify FDMEE seeded objects imported from the FDMEE repository. Instead, copy needed objects, use the " Custom" suffix in names to identify them, and then modify them.

For example, copy the "Open Interface Balances" package as "Open Interface Balances Custom," and make any edits to it.
7. Export customized objects (for example, Project, Packages, Interfaces) from the custom repository and import them into the FDMEE repository.
8. Using the FDMEE Source Adapter user interface, copy the **Open Interface Adapter** to a new adapter and complete the following:
 - For the new adapter, provide a new adapter key. The adapter key can be ten characters long. Once it has been created, it cannot be modified.
 - Make any necessary modifications to the Display Names. Display Name are shown in the Import Format and Data Load Workbench.
 - Specify the ODI Package Name and ODI Project Code as specified in the custom project.

Migrating the Customization

To migrate a customization from a test or production repository:

1. Export the model and project from the custom repository.
2. Import the model and project into the target Oracle Hyperion Financial Data Quality Management, Enterprise Edition repository (using the Synonym Mode INSERT_UPDATE as the Import Type).

PeopleSoft Commitment Control

PeopleSoft's Commitment Control is a budgetary control feature in the PeopleSoft General Ledger product, which supports the posting of budgets and tests transactions against budgetary balances according to the rules configured by users. Using Commitment Control, you can define an alternate structure from the General Ledger to manage budgets based on the Chartfield (account segments) and calendars. For example, you can choose to control budgets at a Division level by Quarter whereas actual expenses are recorded at the Cost Center level by month.

Additionally, you can make large scale changes to budgets, and write revisions as journals back to source systems for posting while maintaining distinctions in PeopleSoft budgets between original proposals, adjustments, revisions, and adjusted budgets. Oracle Hyperion Planning, for example, can be used to prepare the initial budget at the start of the fiscal year. It can also be used to make revisions to the budgets throughout the fiscal year. As budgets are prepared and revised, they have to be validated and posted to Commitment Control, which manages and controls the actual revenue and expenses.

Oracle Hyperion Financial Data Quality Management, Enterprise Edition integrates with Commitment Control by enabling you to use Hyperion Planning to prepare and maintain the budgets. The integration involves:

- loading the Actual from Commitment Control
- validating budgets during preparation and revision against Commitment Control budget definitions
- posting the initial budget to Commitment Control
- posting ongoing budget revisions to Commitment Control

To use Commitment Control within the context of FDMEE, complete the following:

1. In **Source System**, register your General Ledger and HRMS source systems.
2. In **Source System**, to use PeopleSoft Commitment Control, select **Enable Commitment Control**.
3. In **Target Application**, register the target application.
4. In **Import Format**, define an import format that specifies how to map PeopleSoft chart fields to dimensions in the Public Sector Planning and Budgeting applications.
5. In **Data Load Rule**, define a location that identifies the PeopleSoft accounting entity (business unit) from which to load data.
6. In **Data Load Rule**, specify a period mapping type of "Budget."
7. In **Data Load Rule** define global, application, and source mappings that specify how period keys in the PeopleSoft calendar and time periods correspond to periods in your Public Sector Planning and Budgeting budget application such as months, quarters, and years. Options include:
 - Select an "As of Date."

The “As of Date” to determine effective dated records, for example, Budget Definition.

- Select a "Target for Blank Period" if the budget is based on a project period.
- Optionally, in **Period Mappings**, map a budget period by mapping a calendar and period to an FDMEE period.

Commitment Control enables different calendars to be used for different rules. The calendar can be of different granularity and duration. For example, you can map the Chartfield used for rule ranges to the Hyperion Planning Entity dimension. When you specify an explicit period mapping, you can map a Budget period by pointing to a calendar and period to an FDMEE period. The same FDMEE period can be mapped to multiple source periods from different calendars.

8. Run the data load rules.
9. In **Import Format** define an import format that contains write back mappings that identify the Public Sector Planning and Budgeting data to write to the PeopleSoft accounting entity chart fields.
10. In **Data Load Rule**, define a data load rule to write back.

Write-back period mapping is based on explicit period maps. Every Entity is associated with a different Budget Calendar.
11. Run the data load rule to write.

Report Query Table Reference

When creating reports, you can use a base SQL query and a parameter SQL query to enhance the data shown in the report. The base SQL query can be used to pull data from various tables and display the data as report output. For example, using one query definition, you can show different columns or groupings. In one report, you can list amounts by account and group by entity, and in another list amount by entity and group by account.

The parameter SQL query enables you to run a query against the parameters in the report definition. For example, the query can be used to select the Location, Period, Category, or Account groupings. In one report, you can list amounts by account and group by entity, and in another list amount by entity and group by account.

The Oracle Hyperion Financial Data Quality Management, Enterprise Edition tables that are used in the base and parameter SQL query are:

- TDATESEG
- TLOGPROCESS

TDATESEG Table Reference

The TDATESEG table is used to store the data loaded by the user, and the transformation between the source dimension members and results of the mapping process.

Note:

When loading text, the column in TDATESEG it is loaded to is DATA, and the mapped result is loaded to DATAX.

Table I-1 TDATESEG Table Reference

Column Name	Definition	Description
DATAKEY	NUMBER(31,0) NOT NULL ENABLE	System generated unique key for each row of data
PARTIONNKEY	NUMBER(10,0) NOT NULL ENABLE	Location key. Join to TPOVPARTITION to retrieve location information.
CATKEY	NUMBER(10,0) NOT NULL ENABLE	Category Key. Join to TPOVCATEGORY to retrieve category information.
PERIODKEY	DATE NOT NULL ENABLE	Period Key. Join to TPOVPERIOD to retrieve Oracle Hyperion Financial Data Quality Management, Enterprise Edition to EPM period mapping details.

Table I-1 (Cont.) TDATESEG Table Reference

Column Name	Definition	Description
DATAVIEW	VARCHAR2(8 CHAR) DEFAULT 'YTD' NOT NULL ENABLE	Hard coded to YTD for file, and set to YTD for balance sheet and PTD for income statement when pulling data from an Enterprise Resource Planning (ERP) system.
CURKEY	VARCHAR2(10 CHAR) DEFAULT	Currency code of the data.
CALCACCTTYPE	NUMBER(6,0) DEFAULT 9 NOT NULL ENABLE	Indicates if row was imported from source or computed by Logic Group: • 9=Imported • 5=Calculated and Exported • 1=Calculated, and Not Exported
CHANGESIGN	NUMBER(1,0) DEFAULT 0 NOT NULL ENABLE	Indicates that the sign of the imported amount should be reversed: • 0=No Change • 1=Reverse Sign
JOURNALID	VARCHAR2(10 CHAR) DEFAULT	ID for the Journal. User provided value
AMOUNT	NUMBER(29,12) DEFAULT 0 NOT NULL ENABLE	Amount loaded from source
AMOUNTX	NUMBER(29 12) DEFAULT 0 NOT NULL ENABLE	Amount after any transformation rules. This value is loaded to the target application.
DESC1	VARCHAR2(240 CHAR) DEFAULT	Description can be imported from file
DESC2	VARCHAR2(75 CHAR) DEFAULT	Description can be imported from file
ACCOUNT	VARCHAR2(75 CHAR) NOT NULL ENABLE	Account member from source
ACCOUNTX	VARCHAR2(4000 CHAR) DEFAULT	Account member after mapping rules processed
ACCOUNTR	NUMBER(10,0) DEFAULT 0 NOT NULL ENABLE	Key to Mapping used for this dimension. Refers to DATAKEY in TDATEMAPSEG.
ACCOUNTF	NUMBER(6,0) DEFAULT 0 NOT NULL ENABLE	Map types: • 1=Exception • 3=Between • 4=Range
ENTITY	VARCHAR2(75 CHAR) DEFAULT	Entity member from source
ENTITYX	VARCHAR2(75 CHAR) DEFAULT	Entity member after mapping rules processed. This value is exported.
ENTITYR	NUMBER(10,0) DEFAULT 0 NOT NULL ENABLE	Key to Mapping used for this dimension. Refers to DATAKEY in TDATEMAPSEG
ENTITYF	NUMBER(6,0) DEFAULT 0 NOT NULL ENABLE	Map types: • 1=Exception • 3=Between • 4=Range
ICP	VARCHAR2(75 CHAR) DEFAULT	ICP from source
ICPX	VARCHAR2(75 CHAR) DEFAULT	ICP after mapping rules processed. This value is exported.

Table I-1 (Cont.) TDATESEG Table Reference

Column Name	Definition	Description
ICPR	NUMBER(10,0) DEFAULT 0 NOT NULL ENABLE	Key to mapping used for this dimension. Refers to DATAKEY in TDATAMAPSEG.
ICPF	NUMBER(6,0) DEFAULT 0 NOT NULL ENABLE	Map type: • 1=Exception • 3=Between • 4=Range
UD1	VARCHAR2(75 CHAR) DEFAULT	UD1 from source
UD2	VARCHAR2(75 CHAR) DEFAULT	UD2 from source
UD3	VARCHAR2(75 CHAR) DEFAULT	UD3 from source
UD4	VARCHAR2(75 CHAR) DEFAULT	UD4 from source
UD5	VARCHAR2(75 CHAR) DEFAULT	UD5 from source
UD6	VARCHAR2(75 CHAR) DEFAULT	UD6 from source
UD7	VARCHAR2(75 CHAR) DEFAULT	UD7 from source
UD8	VARCHAR2(75 CHAR) DEFAULT	UD8 from source
UD9	VARCHAR2(75 CHAR) DEFAULT	UD9 from source
UD10	VARCHAR2(75 CHAR) DEFAULT	UD10 from source
UD11	VARCHAR2(75 CHAR) DEFAULT	UD11 from source
UD12	VARCHAR2(75 CHAR) DEFAULT	UD12 from source
UD13	VARCHAR2(75 CHAR) DEFAULT	UD13 from source
UD14	VARCHAR2(75 CHAR) DEFAULT	UD14 from source
UD15	VARCHAR2(75 CHAR) DEFAULT	UD15 from source
UD16	VARCHAR2(75 CHAR) DEFAULT	UD16 from source
UD17	VARCHAR2(75 CHAR) DEFAULT	UD17 from source
UD18	VARCHAR2(75 CHAR) DEFAULT	UD18 from source
UD19	VARCHAR2(75 CHAR) DEFAULT	UD19 from source
UD20	VARCHAR2(75 CHAR) DEFAULT	UD20 from source
UD1X	VARCHAR2(75 CHAR) DEFAULT	UD1 after mapping rules processed. This value is exported.
UD1R	NUMBER(10,0) DEFAULT 0 NOT NULL ENABLE	Key to Mapping used for this dimension. Refers to DATAKEY in TDATAMAPSEG
UD1F	NUMBER(6,0) DEFAULT 0 NOT NULL ENABLE	Map type: • 1=Exception • 3=Between • 4=Range
ARCHIVEID	NUMBER(31,0) DEFAULT 0 NOT NULL ENABLE	Future use
HASMEMOITEM	NUMBER(1,0) DEFAULT 0 NOT NULL ENABLE	Future use
STATICDATAKEY	NUMBER(31,0) DEFAULT 0 NOT NULL ENABLE	Future use
ATTR1	VARCHAR2(20 CHAR) DEFAULT	User defined attribute - used as needed for mapping or drill-through
ATTR2	VARCHAR2(20 CHAR) DEFAULT	User defined attribute - used as needed for mapping or drill-through

Table I-1 (Cont.) TDATESEG Table Reference

Column Name	Definition	Description
ATTR3	VARCHAR2(20 CHAR) DEFAULT	User defined attribute - used as needed for mapping or drill-through
ATTR4	VARCHAR2(20 CHAR) DEFAULT	User defined attribute - used as needed for mapping or drill-through
ATTR5	VARCHAR2(20 CHAR) DEFAULT	User defined attribute - used as needed for mapping or drill-through
ATTR6	VARCHAR2(20 CHAR) DEFAULT	User defined attribute - used as needed for mapping or drill-through
ATTR7	VARCHAR2(20 CHAR) DEFAULT	User defined attribute - used as needed for mapping or drill-through
ATTR8	VARCHAR2(20 CHAR) DEFAULT	User defined attribute - used as needed for mapping or drill-through
ATTR9	VARCHAR2(20 CHAR) DEFAULT	User defined attribute - used as needed for mapping or drill-through
ATTR10	VARCHAR2(20 CHAR) DEFAULT	User defined attribute - used as needed for mapping or drill-through
ATTR11	VARCHAR2(20 CHAR) DEFAULT	User defined attribute - used as needed for mapping or drill-through
ATTR12	VARCHAR2(20 CHAR) DEFAULT	User defined attribute - used as needed for mapping or drill-through
ATTR13	VARCHAR2(20 CHAR) DEFAULT	User defined attribute - used as needed for mapping or drill-through
ATTR14	VARCHAR2(20 CHAR) DEFAULT	User defined attribute - used as needed for mapping or drill-through
CODE_COMBINATION_ID	VARCHAR2(155 CHAR)	Used for integration with Oracle E-Business Suite.
AMOUNT_YTD	NUMBER(29,12)	YTD Amount. Used for E-Business Suite, Peoplesoft, Fusion data sources
AMOUNT_PTD	NUMBER(29,12)	PTD Amount. Used for E-Business Suite, Peoplesoft, Fusion data sources
LOADID	NUMBER(15,0)	Process ID that created or updated this row.
RULE_ID	NUMBER(15,0)	Data Rule ID used to create this row. Join to AIF_BALANCE_RULES for details.
STAT_BALANCE_FLAG	VARCHAR2(1 CHAR)	Indicates if balance is a statistic: • Y=Stat • N=Balance
VALID_FLAG	VARCHAR2(1 CHAR)	Indicates if row has valid mappings: • Y=Valid • N=Not Valid • I=Ignore

TLOGPROCESS Table Reference

The TLOGPROCESS table is used to store the workflow process status for a location, category, and period.

Table I-2 TLOGPROCESS Table Reference

Column Name	Definition	Description
PARTITIONKEY	NUMBER(10,0) NOT NULL ENABLE	Location key. Join to TPOVPARTITION to retrieve location information.
CATKEY	NUMBER(10,0) NOT NULL ENABLE	Category Key. Join to TPOVCATEGORY to retrieve category information.
PERIODKEY	DATE NOT NULL ENABLE	Period Key. Join to TPOVPERIOD to retrieve Oracle Hyperion Financial Data Quality Management, Enterprise Edition to EPM period mapping details.
RULE_ID	NUMBER(15,0) NOT NULL ENABLE	Data Rule ID. Join to AIF_BALANCE_RULES for details.
PROCESSIMP	NUMBER(1,0) DEFAULT 0 NOT NULL ENABLE	Status for Import step: • 0=Not started or failed • 1=Successful
PROCESSIMPNOTE	VARCHAR2(50 CHAR) DEFAULT NULL	Textual note on Validate status: • Import Successful • Recalculated OK • Import Error • Recalculate Failed • MultiLoad • BypassDataLoad • Logic Calc Err • Map Calc Err
PROCESSVAL	NUMBER(1,0) DEFAULT 0 NOT NULL ENABLE,	Status for Validate step: • 0=Not started or failed • 1=Successful
PROCESSVALNOTE	VARCHAR2(50 CHAR) DEFAULT NULL,	Textual note on Validate step: • Validate Successful • Error= x records (Where X = how many members did not have map rules) • BypassDataLoad
PROCESSEX	NUMBER(1,0) DEFAULT 0 NOT NULL ENABLE,	Status for Export step: • 0=Not started or failed • 1=Successful
PROCESSEXNOTE	VARCHAR2(50 CHAR) DEFAULT NULL	Textual note on Export step: • Last successful export • Export -B Successful • Export Successful • BypassDataLoad

Table I-2 (Cont.) TLOGPROCESS Table Reference

Column Name	Definition	Description
PROCESSENTLOAD	NUMBER(1,0) DEFAULT 0 NOT NULL ENABLE	Status for Load step: • 0=Not started or failed • 1=Successful
PROCESSENTLOADNOTE	VARCHAR2(50 CHAR) DEFAULT NULL	Textual note on Load status: • Time Date stamp for success • Load Error • BypassDataLoad
PROCESSENTVAL	NUMBER(1,0) DEFAULT 0 NOT NULL ENABLE,	Status for Check step: • 0=Not started or failed • 1=Successful
PROCESSENTVALNOTE	VARCHAR2(50 CHAR) DEFAULT NULL	Textual note on Check step: • Check Successful • Check Error • BypassDataLoad
PROCESSCERT	NUMBER(1,0) DEFAULT 0 NOT NULL ENABLE,	Status for Certification step: • 0=Not started or unsubmitted • 1=Submitted
PROCESSCERTNOTE	VARCHAR2(50 CHAR) DEFAULT NULL,	Textual note on Load status: • Controls Submitted • Controls Cleared • Not Submitted • No Controls Found for Profile • No Controls Group Assigned • Error Setting Status
PROCESSASSES	NUMBER(1,0) DEFAULT 0 NOT NULL ENABLE,	Status for Assessment (process explorer) step: • 0=Not started or unsubmitted • 1=Submitted
PROCESSASSESNOTE	VARCHAR2(50 CHAR) DEFAULT NULL,	Textual note on Load status: • Controls Submitted • Controls Cleared • Not Submitted • No Controls Found for Profile • No Controls Group Assigned • Error Setting Status
PROCESSCHILDDONE	NUMBER(1,0) DEFAULT 0 NOT NULL ENABLE	Status for Certification status for parent locations step: • 0=Not started or all children not complete • 1=All children complete
PROCESSCHILDDONENOTE	VARCHAR2(50 CHAR) DEFAULT NULL	Textual note on Certification status for parent location: • Children Submitted • No Children
PROCESSUD1	NUMBER(1,0) DEFAULT 0 NOT NULL ENABLE	Not used
PROCESSUD1NOTE	VARCHAR2(50 CHAR) DEFAULT NULL	Not used

Table I-2 (Cont.) TLOGPROCESS Table Reference

Column Name	Definition	Description
PROCESSUD2	NUMBER(1,0) DEFAULT 0 NOT NULL ENABLE	Not used
PROCESSUD2NOTE	VARCHAR2(50 CHAR) DEFAULT NULL	Not used
PROCESSUD3	NUMBER(1,0) DEFAULT 0 NOT NULL ENABLE	Not used
PROCESSUD3NOTE	VARCHAR2(50 CHAR) DEFAULT NULL	Not used
PROCESSUD4	NUMBER(1,0) DEFAULT 0 NOT NULL ENABLE	Not used
PROCESSUD4NOTE	VARCHAR2(50 CHAR) DEFAULT NULL	Not used
PROCESSENDTIME	DATE DEFAULT TO_DATE('01/01/1900', 'MM/DD/YYYY') NOT NULL ENABLE	Last update time/date
BLNWCDIRTY	NUMBER(1,0) DEFAULT 0 NOT NULL ENABLE	Flag used to indicate that maps (WC = WildCard) must be recalculated prior to validating: 0=OK 1=Re-calculate location Map rules have changed after data was imported. This causes the Calculate flag to be displayed
BLNLOGICDIRTY	NUMBER(1,0) DEFAULT 0 NOT NULL ENABLE	Flag used to indicate the LOGIC must be recalculated prior to validating
BLNVALDIRTY	NUMBER(1,0) DEFAULT 1 NOT NULL ENABLE	Flag used to indicate when Validation workflow must be re-run: 0=OK 1=re-process validations Map rules have changed after data was imported. This causes the Calculate flag to be displayed.
INTLOCKSTATE	NUMBER(6,0) DEFAULT 50 NOT NULL ENABLE	Location POV lock status: 50=open 60=locked
PROCESSSTATUS	NUMBER(10,0) DEFAULT 0 NOT NULL ENABLE	Current state of workflow for location/category/per. Valid status from tLogProcessStates

J

System Maintenance Tasks

You can run system processes to maintain and cleanup all runtime artifacts, such as the Process tables, Staging tables or Inbox / Outbox folders. Often the tables and folders contain vast amounts of data, which you may no longer need. With the System Maintenance Tasks feature, you can purge standard tables and folder by scheduling system processes and executing them.

Note:

All applications *not* assigned to a folder are purged when a single application is selected for a purge. The default application folder is generic and the purge script focuses on the folder in which the selected application resides. In this case if you want to prevent an application from being purged, save it to an independent folder

To facilitate the use of the Purge Scripts, Oracle Hyperion Financial Data Quality Management, Enterprise Edition provides the following:

- A set of custom scripts is shipped to the bin/system directory.
The scripts include:
 - Maintain Application Folder
 - Maintain Process Table
 - Maintain EBS GL Balances Table
 - Maintain PeopleSoft Ledger Table
 - Maintain FDMEE Data Tables
- Scripts are registered as system scripts in script registration.
- Script are registered as part of installation with QUERYID = 0 and APPLICATIONID = 0.
- The script group "System" is created and system scripts are assigned to it.
- Script execution displays system scripts when the user has access irrespective of the target application in the POV.
- You can run purge scripts from the Script Execution screen.
- The ODI process executes the scripts from the bin/system dir instead of the data/scripts/custom directory.

Maintain Application Folder

The Maintain Application Folder process purges files from the inbox, outbox, and data folder directories. Oracle Hyperion Financial Data Quality Management, Enterprise

Edition accepts a separate Days to Keep parameter for each of the folders. If the value is not specified for a specific folder, FDMEE skips the folder.

In addition, FDMEE checks the `inbox` and `outbox` subdirectories under the respective folders and deletes any files. In the `data` subdirectory, FDMEE skips the `scripts` directory because it holds customer scripts.

Maintain Process Table

This process maintains the following execution tables:

- AIF_PROCESSES
- AIF_PROCESS_DETAILS
- AIF_PROCESS_LOGS
- AIF_PROCESS_PARAMETERS
- AIF_PROCESS_PERIODS
- AIF_PROCESS_STEPS
- AIF_BAL_RULE_LOADS
- AIF_BAL_RULE_LOAD_PARAMS
- AIF_BATCH_JOBS
- AIF_BATCH_LOAD_AUDIT
- AIF_TEMP

It accepts the number of days to keep as a parameter.

Maintain EBS GL Balances Table

This process maintains the `AIF_EBS_GL_BALANCES_STG` table. EBS General Ledger balances contain a snapshot of the General Ledger balances and are deleted based on a General Ledger period.

The process determines the list of General Ledger periods between the start and end periods.

The parameter for this process are:

- Source System
- Start Period
- End Period

Maintain PeopleSoft Ledger Table

This process maintains the `AIF_PS_LEDGER_STG` table. PeopleSoft General Ledger balances contain a snapshot of the General Ledger balances and are deleted based on General Ledger period.

The process determines the list of General Ledger periods between the start and end periods.

The parameter for this process are:

- Source System
- Start Period
- End Period

Maintain FDMEE Data Tables

The Oracle Hyperion Financial Data Quality Management, Enterprise Edition data tables store archives of exported data for audit and drill down purposes. These tables can grow over period of time and you can purge these tables as needed. The maintenance process deletes the following tables:

- TDATAMAPSEG
- TDATASEG
- TPROCESSLOG

The parameters are:

- Target Application
- Category
- Start Period
- End Period

Executing Purge Scripts

To execute a purge script:

1. On the **Workflow** tab, under **Other**, select **Script Execution**.
2. In **Script Execution**, and then in **Custom Script Group**, select **System Maintenance Task**.
3. From the **Scripts** grid, select the purge script.
4. Click **Execute**.
5. When prompted, enter parameter values on the Execute Script screen.
For example, you might select the target application, Start Period, or End Period.
6. **Optional**, click **Schedule**.
For information on scheduling jobs, see [Scheduling Jobs](#).
7. From **Execution Mode**, select the online method of running the report.
The online method processes the report immediately.
8. Click **OK**.

K

Setting up Jython, Eclipse and Python

This appendix explains how to set up Jython, Eclipse and PyDev for use with the Oracle Hyperion Financial Data Quality Management, Enterprise Edition scripting component.

This section is specific to the 11.1.2.3.100 release of FDMEE. Because Jython, Eclipse and Python are not Oracle products, always refer to these products' documentation for the latest updates and changes. Also see [PyDev manual](#).

To set up Jython:

1. Create a working directory.

For example, create: C:\FDMEE.

2. Download Jython to the working directory.

The download is available at [Jython download](#) site.

Double-click the Jython installer jar and select the following options:

- English Language
- Standard Install
- Default Location (C:\FDMEE\jython2.5.1)
- Current Java Home

3. Download **Eclipse** to working directory and extract.

The Eclipse download is available at <http://www.eclipse.org/downloads>



Note:

In addition to Eclipse, users may use Notepad++ with the Jython add-in, or the [Python Fiddle](#) site to write and test scripts. Pythonfiddle is a good site for writing event scripts, and the Chrome browser is recommended for pythonfiddle. For more advanced scripts, Eclipse or Jdeveloper are recommended

4. Launch **Eclipse** from the C:\FDMEE\eclipse\eclipse.exe.

5. Select the directory and select a default location for your Workspace.

For example select c:\FDMEE\Workspace

6. If the Welcome page is displayed, skip it by selecting the Workbench link in the top right corner.

7. Select the following menu option to update your preferences:

- a. Select **Windows**, and then select **Preferences**.
- b. Add **PyDev** to the Available Software Sites:

To do this:

- i. Select **Install/Update**, then select **Available Software Sites**, and then click **Add**.
- ii. In **Name**, enter **PyDev**.
- iii. In **Location**, enter: [PyDev](#) site.

8. Change the default text editor options:

To do this:

- a. Select **General**, then select **Editors**, and then select **Text Editors**.
- b. For the **Displayed tab width**, enter **2**.
- c. Enable the **Insert spaces for tabs** option.
- d. Enable the **Show print margin** option.
- e. Enable the **Show line numbers** option.

9. Select the **Menu** option.

To do this:

- a. Select **Help**, and then select **Install New Software...**
- b. In **Work with**, select **PyDev**.
- c. From **PYDev** available software list, select the **PyDev** node, and then click **Next**.
- d. Confirm the PyDev for Eclipse installation, and then click **Next**.
- e. Accept the licensing terms, and then click **Finish**.
- f. Once the install starts, you are prompted about Aptana PyDev; and PyDev; Aptana certificates. Make sure you select the certificates, and then click **OK**.
- g. Restart **Eclipse** to complete the PyDev installation.

10. Update your PyDev preferences:

To do this:

- a. Select **Windows**, and then select **Preferences**.
- b. Configure the Jython interpreter used by PyDev:
 - i. Select **PyDev**, and then select **Interpreter-Jython**.
 - ii. Click **New**.
 - iii. Click **Browse**.
 - iv. Select `C:\FDME\jython2.5.1\jython.jar`
 - v. Click **OK**.
 - vi. When prompted to select the folders to add to the SYSTEM python path, do not change any selection, and then click **OK**.
 - vii. Click **OK** to close the Preferences window.