

Oracle® Enterprise Performance Foundation

User's Guide

Release 11*i*

Part No. B14460-03

July 2006

Primary Author: Greg Becker

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software-Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments

Preface

1 Oracle Enterprise Performance Foundation Basics

What Is Oracle Enterprise Performance Foundation?	1-1
Data and Metadata Organization	1-1
Core Concepts: Value Sets, Global Value Set Combinations, and Ledgers	1-2
Shared Features	1-3
The Home Page	1-3
User Responsibilities	1-4
Application Preferences	1-4

2 Common Rule Management Tasks

Overview of Common Rule Management Tasks	2-1
The Rule Home Page	2-2
Searching for Rules	2-4
Creating Rules	2-4
Viewing and Updating Rules	2-5
Duplicating Rules	2-6
Deleting Rules	2-7
Exporting, Importing, and Migrating Rules.	2-7

3 Dimension and Hierarchy Management

About Dimension and Hierarchy Management	3-1
Working with Dimension Definitions	3-1
Working with Dimension Attributes	3-2
Working with Dimension Levels	3-5
Working with Members	3-7
Working with Hierarchies	3-11
Viewing Dimension Defaults.	3-22

4 Using Conditions to Select Data

About Conditions	4-1
----------------------------	-----

Working with conditions	4-1
Creating Conditions	4-4
Working with Dimension Components	4-4
Working with Data Components	4-7

5 Using the Data Inspector to View and Edit Tables

About the Data Inspector and Data Inspector Rules	5-1
Creating Data Inspector Rules	5-2
Viewing Data Inspector Rules	5-3
Running Data Inspector Rules	5-3
Duplicating Data Inspector Rules	5-4
Updating Data Inspector Rule Names and Descriptions	5-4
Updating Selection Settings for Data Inspector Rules	5-4
Deleting Data Inspector Rules	5-5

6 Working with Rule Sets

About Rule Sets	6-1
Creating Rule Sets	6-1
Viewing Rule Sets	6-2
Running Rule Sets	6-2
Duplicating Rule Sets	6-3
Updating Rule Set Names, Descriptions, and Access Settings	6-3
Updating Rule Set Definition Settings	6-3
Deleting Rule Sets	6-4

7 Working with Data Set Groups

About Data Set Groups	7-1
Creating Data Set Groups	7-2
Viewing Data Set Groups	7-3
Duplicating Data Set Groups	7-3
Updating Data Set Groups	7-4
Deleting Data Set Groups	7-5

8 Working with Requests

About Concurrent Programs and Requests	8-1
Submitting Requests	8-1
Programs	8-3
Viewing Requests	8-9

9 Removing Rule Execution Results

About Rule Execution Results and Process Locking	9-1
Viewing Rule Execution Results	9-2
Removing Rule Execution Results and Process Locks	9-2

10	Performing Administrative Tasks	
	About Administrative Tasks in Enterprise Performance Foundation	10-1
	Performing Table and Column Registration Tasks	10-1
	Performing Dimension Administration Tasks	10-6
	Working with Value Sets	10-7
	Working with Global Value Set Combinations	10-8
	Working with Registered Database Links	10-9
	Working with Process Tuning Options.	10-10
	Working with Security Folders	10-13
	Customizing the Home Page	10-15
11	Working with Rule Approval Status	
	About Rule Approval and Production Data Sets	11-1
	The Rule Approval Process	11-2
	The Rule Deletion Process	11-2
12	Implementing Enterprise Performance Foundation with Oracle General Ledger	
	Introduction	12-1
	General Ledger Rules Overview	12-2
	Defining General Ledger Rules	12-10
	Synchronizing General Ledger and Enterprise Performance Foundation.	12-25
	Miscellaneous Notes and Best Practices for Enterprise Performance Foundation and Oracle General Ledger Rules	12-28
13	Functional Dimension Definition	
	Introduction	13-1
	Creating Functional Dimension Definition Rules	13-2
	Searching for Functional Dimension Definition Rules.	13-3
	Updating Functional Dimension Definition Rules	13-3
	Deleting Functional Dimension Definition.	13-4
A	Interface Tables and External Data Loaders	
	Introduction	A-1
	Dimension Member Loader	A-7
	Dimension Hierarchy Loader	A-52
	External General Ledger Data Loader	A-72
	Detail Client Data Loader	A-88
	Dimension Loader.	A-103
	Data Loader.	A-107
B	Profile Options	
	Enterprise Performance Foundation Profile Options.	B-1
	Profile Options in Application Preferences	B-3

C	Using Oracle Enterprise Performance Foundation with Oracle Discoverer	
	Using Oracle Enterprise Performance Foundation with Oracle Discoverer	C-1
D	Business Events	
	Business Events	D-1
E	Integrating Enterprise Performance Foundation with Oracle Warehouse Builder	
	Introduction	E-1
	Installation	E-1
	Integration Components	E-2
	User Procedures	E-7

Index

Send Us Your Comments

Oracle Enterprise Performance Foundation User's Guide, Release 11i

Part No. B14460-03

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document. Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Applications Release Online Documentation CD available on Oracle MetaLink and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: appsdoc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

Preface

Intended Audience

Welcome to Release 11i of the *Oracle Enterprise Performance Foundation User's Guide*.

This guide assumes you have a working knowledge of the following:

- The principles and customary practices of your business area
- Oracle Enterprise Performance Foundation

If you have never used Oracle Enterprise Performance Foundation, Oracle suggests you attend one or more of the Oracle Applications training classes available through Oracle University.

- Oracle Self-Service Web Applications.

To learn more about Oracle Self-Service Web Applications, read the *Oracle Self-Service Web Applications Implementation Manual*.

- The Oracle Applications graphical user interface

To learn more about the Oracle Applications graphical user interface, read the *Oracle Applications User's Guide*.

See Related Information Sources on page x for more Oracle Applications product information.

TTY Access to Oracle Support Services

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

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Structure

1 Oracle Enterprise Performance Foundation Basics

2 Common Rule Management Tasks

This chapter focuses on the rule management tasks that are common across all rules in this application.

3 Dimension and Hierarchy Management

4 Using Conditions to Select Data

5 Using the Data Inspector to View and Edit Tables

6 Working with Rule Sets

7 Working with Data Set Groups

8 Working with Requests

9 Removing Rule Execution Results

10 Performing Administrative Tasks

11 Working with Rule Approval Status

12 Implementing Enterprise Performance Foundation with Oracle General Ledger

13 Functional Dimension Definition

A Interface Tables and External Data Loaders

B Profile Options

C Using Oracle Enterprise Performance Foundation with Oracle Discoverer

D Business Events

E Integrating Enterprise Performance Foundation with Oracle Warehouse Builder

Related Information Sources

Oracle Applications User's Guide

This guide explains how to enter data, query, run reports, and navigate using the graphical user interface (GUI). This guide also includes information on setting user profiles, as well as running and reviewing reports and concurrent processes.

You can access this user's guide online by choosing "Getting Started with Oracle Applications" from any Oracle Applications help file.

Oracle Applications Concepts

This guide provides an introduction to the concepts, features, technology stack, architecture, and terminology for Oracle Applications Release 11i. It provides a

useful first book to read before an installation of Oracle Applications. This guide also introduces the concepts behind Applications-wide features such as Business Intelligence (BIS), languages and character sets, and Self-Service Web Applications.

Installing Oracle Applications

This guide provides instructions for managing the installation of Oracle Applications products. In Release 11i, much of the installation process is handled using Oracle Rapid Install, which minimizes the time to install Oracle Applications and the Oracle technology stack by automating many of the required steps. This guide contains instructions for using Oracle Rapid Install and lists the tasks you need to perform to finish your installation. You should use this guide in conjunction with individual product user guides and implementation guides.

Upgrading Oracle Applications

Refer to this guide if you are upgrading your Oracle Applications Release 10.7 or Release 11.0 products to Release 11i. This guide describes the upgrade process and lists database and product-specific upgrade tasks. You must be either at Release 10.7 (NCA, SmartClient, or character mode) or Release 11.0, to upgrade to Release 11i. You cannot upgrade to Release 11i directly from releases prior to 10.7.

About Document

For information about implementation and user documentation, instructions for applying patches, new and changed setup steps, and descriptions of software updates, refer to the About Document for your product. About Documents are available on OracleMetaLink for most products starting with Release 11.5.8.

Maintaining Oracle Applications

Use this guide to help you run the various AD utilities, such as AutoUpgrade, Auto Patch, AD Administration, AD Controller, AD Relink, License Manager, and others. It contains how-to steps, screenshots, and other information that you need to run the AD utilities. This guide also provides information on maintaining the Oracle applications file system and database.

Oracle Applications System Administrator's Guide

This guide provides planning and reference information for the Oracle Applications System Administrator. It contains information on how to define security, customize menus and online help, and manage concurrent processing.

Oracle Alert User's Guide

This guide explains how to define periodic and event alerts to monitor the status of your Oracle Applications data.

Oracle Applications Developer's Guide

This guide contains the coding standards followed by the Oracle Applications development staff and describes the Oracle Application Object Library components that are needed to implement the Oracle Applications user interface described in the *Oracle*

Applications User Interface Standards for Forms-Based Products. This manual also provides information to help you build your custom Oracle Forms Developer forms so that the forms integrate with Oracle Applications.

Oracle Applications Product Update Notes

Use this guide as a reference for upgrading an installation of Oracle Applications. It provides a history of the changes to individual Oracle Applications products between Release 11.0 and Release 11i. It includes new features, enhancements, and changes made to database objects, profile options, and seed data for this interval.

Oracle Workflow Administrator's Guide

This guide explains how to complete the setup steps necessary for any Oracle Applications product that includes workflow-enabled processes, as well as how to monitor the progress of runtime workflow processes.

Oracle Workflow Developer's Guide

This guide explains how to define new workflow business processes and customize existing Oracle Applications-embedded workflow processes. It also describes how to define and customize business events and event subscriptions.

Oracle Workflow User's Guide

This guide describes how Oracle Applications users can view and respond to workflow notifications and monitor the progress of their workflow processes.

Oracle Workflow API Reference

This guide describes the APIs provided for developers and administrators to access Oracle Workflow.

Oracle eTechnical Reference Manuals

Each eTechnical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for a specific Oracle Applications product. This information helps you convert data from your existing applications, integrate Oracle Applications data with non-Oracle applications, and write custom reports for Oracle Applications products. Oracle eTRM is available on OracleMetalink

Oracle Applications Message Manual

This manual describes all Oracle Applications messages. This manual is available in HTML format on the documentation CD-ROM for Release 11i.

Do Not Use Database Tools to Modify Oracle Applications Data

Oracle **STRONGLY RECOMMENDS** that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications data unless otherwise instructed.

Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus 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, 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 to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. 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.

Oracle Enterprise Performance Foundation Basics

This chapter covers the following topics:

- What Is Oracle Enterprise Performance Foundation?
- Data and Metadata Organization
- Core Concepts: Value Sets, Global Value Set Combinations, and Ledgers
- Shared Features
- The Home Page
- User Responsibilities
- Application Preferences

What Is Oracle Enterprise Performance Foundation?

Oracle Enterprise Performance Foundation is an open and shared repository of data and business rules that provides the framework for all of the applications in the Corporate Performance Management set of products. This single source of truth provides a set of features that allow you to manage repository metadata and enable you to generate management reports and perform analyses.

Enterprise Performance Foundation is tightly integrated with the E-Business Suite and Oracle General Ledger, but data from external systems is just as easily integrated and utilized within Oracle Enterprise Performance Foundation.

Data and Metadata Organization

Enterprise Performance Foundation is fully metadata-driven, and is comprised of dimensions, dimension hierarchies, fact tables, business-specific metadata, and business rules. In addition, the architecture uses full multiple language support (MLS) for use with the standard set of E-Business Suite localizations.

The Enterprise Performance Foundation package includes the data model, external loaders, and direct integration with the E-Business Suite.

Dimensions and dimension members

A *dimension* categorizes data into a single object. Examples of dimensions are Time, Geography, Organization, and Product. *Dimension members* are the individual

items that make up a dimension. For example, San Francisco and New York might be members of an Organization dimension.

Hierarchies and levels

Dimension members may be arranged in levels, with each level representing the aggregated total of the data from the level below. For example, an Organization hierarchy might consist of levels for World > Continent > Country > Site > Cost Center — each aggregating up to the parent above. Members can belong to specific levels: for example, Europe would be a continent, France would be a country, Lyons a site, and so forth.

Hierarchies can specify the levels through which data is aggregated and allocated. When users view data, they can drill up and down a hierarchy to see data at various levels. Users can also select data by level.

A dimension can have multiple hierarchies; for example, Organization might have both a geographic hierarchy such as World > Continent > Country > Site > Cost Center and a managerially driven hierarchy such as Total World > Lines of Business > Cost Center, where each hierarchy is derived from the same organization domain.

Attributes

An attribute is a property or qualifier that describes a dimension member. An attribute may be a date, a number, or a character string. For example, the Geography dimension may have a Population attribute that designates how many people live in that area.

Fact tables

The system is comprised of fact tables that are defined by the dimensions present in the metadata. Users may use the pre-seeded fact tables, add dimensions to the existing tables or create their own new tables.

Business-specific metadata

Business-specific metadata comes pre-seeded with each of the specialized applications that use Enterprise Performance Foundation, and is thus included with those applications rather than with Enterprise Performance Foundation. However, Enterprise Performance Foundation is easily extended by these other applications.

Business rules

Enterprise Performance Foundation supports a framework for application-specific business rules, in addition to some pre-seeded generic business rules. For example, the Oracle Financial Consolidation Hub application provides business rules specific to the functions related to the consolidation of data across multiple sets of books. In addition, Enterprise Performance Foundation includes shared business rules for data conditions which act as filters, and a data inspector to view data in the fact tables within the system.

Core Concepts: Value Sets, Global Value Set Combinations, and Ledgers

The organization of metadata within Enterprise Performance Foundation is similar to the way that value sets, charts of accounts and sets of books are defined within the Oracle

E-Business Suite. However, this base definition may be easily extended to meet each organization's specific business requirements.

About value sets, global value set combinations, and ledgers

The core definition of data in the fact tables in the Enterprise Performance Foundation data model is by the ledger dimension. In turn, the ledger dimension is attributed by a component called a global value set combination, a calendar and a currency. Note that this is similar to a set of books within the E-Business Suite, which is defined by a chart of accounts, a currency and a calendar.

A global value set combination is similar to a chart of accounts, and is comprised of a value set for each dimension within the application. (This is different from the chart of accounts in the E-Business suite, since the chart of accounts is a subset of all "dimensions" within the transactional systems.)

Because a global value set combination defines a value set for each dimension, it effectively determines the domain for each dimension for a selected ledger. When working in an application, the user must select a default ledger; which specifies a global value set combination; the global value set combination, in turn, defines the value set for each dimension for the rules related to a specific ledger and the data dimensioned by that ledger.

Example

For example, suppose that there are two ledgers in your system, Ledger1 and Ledger2.

The global value set combination that is related to Ledger1 is GVSC1. GVSC1, in turn, consists of one value set for each dimension in the system, and the value set for the Line Item dimension in GVSC1 is LineItem1. The LineItem1 value set contains Line Item dimension members LineA, LineB, and LineC.

The global value set combination for Ledger2 is GVSC2. The value set for the Line Item dimension in GVSC2 is LineItem2, which contains Line Item dimension members LineD, LineE, and LineF.

If you create a business rule for Ledger1, the available Line Item dimension members are those in the LineItem1 value set: LineA, LineB, and LineC. Similarly, if you create a rule for Ledger2, the available Line Item members are those in the LineItem2 value set: LineD, LineE, and LineF. You cannot select any of the LineItem2 members for a rule for Ledger1, and you cannot select any of the LineItem1 members for a rule for Ledger2.

Shared Features

All of the features provided by Enterprise Performance Foundation are shared across all Corporate Performance Management applications, as the applications all use the same core data model and fact tables.

The Home Page

The Home page displays the following:

- The name of the system and the user name that you are logged in under.
- An information section, in which you can display information that you might find useful.

- Notifications that have been directed to you.

You can use the View box on the Home page to specify the types of notifications that you want to view. To view and respond to a notification, click on the link for the notification in the Subject column.

- Links to any Web sites that you would like to be able to access directly from the Home page.

For information about customizing the Home page, see *Customizing the Home Page*, page 10-15.

User Responsibilities

The Enterprise Performance Foundation Administrator user responsibility is provided with Enterprise Performance Foundation. This responsibility allows you to perform various system administration tasks, and is the responsibility that must be used when defining and running the business rules that move data, dimensions, and hierarchies from Oracle General Ledger into Enterprise Performance Foundation.

Other applications may provide additional user responsibilities. See the appropriate product documentation for descriptions of those responsibilities.

Application Preferences

The **Application Preferences** link in the global toolbar area allows you to set various defaults specific to your user account. You must specify the default security folder and default ledger before using any applications. The Effective Date user preference, which selects the appropriate version, is required for viewing hierarchies in rules.

Common Rule Management Tasks

This chapter focuses on the rule management tasks that are common across all rules in this application.

This chapter covers the following topics:

- Overview of Common Rule Management Tasks
- The Rule Home Page
- Searching for Rules
- Creating Rules
- Viewing and Updating Rules
- Duplicating Rules
- Deleting Rules
- Exporting, Importing, and Migrating Rules

Overview of Common Rule Management Tasks

The rule management tasks that are common to business rules in this and other applications are as follows.

- The Rule Home Page, page 2-2
- Searching for Rules, page 2-4
- Creating Rules, page 2-4
- Viewing and Updating Rules, page 2-5
- Duplicating Rules, page 2-6
- Deleting Rules, page 2-7
- Exporting, Importing, and Migrating Rules, page 2-7

Note: You can perform these tasks from the home page for the type of rule with which you are working. Depending on the rule type, some tasks might not be available.

The procedures for carrying out these tasks are the same for each rule type, except for rule-specific steps explicitly stated in the rule-specific documentation.

The Rule Home Page

The Rule home page is the gateway to all rules and related functionality of the application. From there, you can navigate to other related pages.

On the header of the Rule home page, you can perform simple queries on Folder, Rule Name and Effective Date.

The following table shows the page components.

Name	Type	Default Value	Required/Optional	Updatable	LOV, additional information
Folder	Drop Down	Set in Application Preferences	Required - for filtering the rules under the folder	No – Only able to select from presented list.	N/A
(Rule) Name	Text Box	None	Optional – for filtering the rules on Rule Name	Yes	You can specify all or part of a rule name For example, if you want to see only those Rules which start with 'A' – Enter A in the text field
Effective Date	Date Field	None	Optional – for filtering the rules on Version Effective Dates	Yes	Returns those Rules with Version effective dates that correspond with the date you enter
Go	Button	N/A	N/A	No	Initiates rule search based on specified criteria.
Create Rule	Button	N/A	N/A	No	Initiate the Data or Ledger Loader rule creation process.
Expand All	Button	N/A	N/A	No	Displays rules with version information
Collapse All	Button	N/A	N/A	No	Displays only with the Rule names
Focus	Button	N/A	N/A	No	Displays the selected Rule's Version information will be shown

Name	Type	Default Value	Required/ Optional	Updatable	LOV, additional information
(Rule) Name	Link	N/A	N/A	No	Links to the Rule's basic details
Start Date	Display Value	N/A	N/A	No	Version effective start date
End Date	Display Value	N/A	N/A	No	Version effective end date
Approval Status	Display Value	N/A	N/A	No	Displays whether this Rule version is approved, submitted, or new
Approved By	Display Value	N/A	N/A	No	Who approved the Rule version
Approved Date	Display Value	N/A	N/A	No	When was the rule version approved
Submit	Button	N/A	N/A	No	Click to have the rule version approved Approved rules can run against production or non-production datasets Unapproved rules can only run against non-production datasets.
Revert	Icon	N/A	N/A	No	
Export	Button	N/A	N/A	N/A	Initiates process for exporting rules
Migrate	Button	N/A	N/A	N/A	Initiates process for migrating rules

Name	Type	Default Value	Required/Optional	Updatable	LOV, additional information
Duplicate	Button	N/A	N/A	N/A	Initiates process for duplicating rules. Explained later in this document
Run	Button	N/A	N/A	N/A	Initiates process for running Rules Explained later in this document.

Searching for Rules

Search for a business rule to perform any of the following tasks:

- Update, duplicate, export, migrate, or delete existing rules or versions
- Create a new version
- Define methodologies for new products

Procedure:

1. Navigate to the rule home page, page 2-2 for the appropriate rule type.
2. Search for the rule, as follows:
 1. Select the folder in which the rule is stored.
 2. (Optional) Enter the name of the rule.
 3. (Optional) Select the effective date.
 4. Click Go.

Only rules that match the search criteria are displayed.

Related Topics

Overview of Common Rule Management Tasks, page 2-1

Creating Rules

You create a rule to specify the way you want a particular task or business process to be carried out by the application. Creating a rule is a two-step process, in which you first specify the properties for the rule itself, and then specify the properties for the rule version.

Procedure to Create a Rule and Version:

1. Navigate to the home page, page 2-2 of the rule you want to create.
2. Click Create to display the rule definition page.

3. Select the folder in which you want to store the rule.
4. Enter a name for the rule.

Important: The name of a rule must be unique across all the rules and rule types in the entire database, not just at the folder level.

5. (Optional) Enter a brief description for the rule.
6. Select the required access for other users.
7. Click Continue.
The version definition page is displayed.
8. Type the name of the version for the rule.
9. Select the effective start and end dates using the date picker. Alternatively, you can type them in the space provided.

Important: Each version must have a unique date range, as compared to all other versions for the same rule.

Note: If not specified otherwise in the related profile option, the defaults for the effective start and end dates are January 1, 1900 and December 31, 2500.

10. Specify any other properties or options that may apply for the version that you are creating.
11. Click Finish.

Related Topics

Overview of Common Rule Management Tasks, page 2-1

Viewing and Updating Rules

You can view existing rules and rule versions, and you can update certain properties for rules and versions.

Note the following considerations with regard to updating rules and versions:

- If you update a rule that has been approved, you must resubmit the rule for approval before it can be run.
- Rules that have been run are process locked, so you cannot update a rule that has been run except by removing the rule execution results and process lock. However, you can create a new (duplicate) version of a rule that has been run and edit then edit the new version without having to undo previous results.

Procedure:

1. Navigate to the home page, page 2-2 of the rule you want to update.
2. Search for a rule. For further information, see Searching for Rules, page 2-4.
3. Click Update corresponding to the rule or version that you want to update if you are familiar with the rule or version details and would like to update the rule or

version directly. Alternatively, click on the rule or version to view details and then click Update on the View page.

Procedure to Update a Rule

1. Update the Name or Description.
2. Click Apply.

Procedure to Update a Version

1. Update the Name.
2. Update the Effective Start Date and the Effective End Date using the date picker. Alternatively, you can enter them in the space provided.

Caution: The updated version's date range must not overlap with any of the existing version date ranges.

3. Click Apply.

Related Topics

Overview of Common Rule Management Tasks, page 2-1

Duplicating Rules

You can duplicate rules and versions to avoid having to enter data multiple times. This saves time and effort and also reduces mistakes. You can duplicate only the version, or you can duplicate both the rule and the version.

When duplicating a version, the rule-related details cannot be updated. All existing versions for a rule are listed at the bottom of the duplicate page.

When you duplicate the version and the rule, a new rule is created with a copy of the version.

Procedure:

1. Navigate to the home page, page 2-2 of the rule or version you want to duplicate.
2. Search for a rule. For further information, see Searching for Rules, page 2-4.
3. Click Duplicate corresponding to the version of the rule that you want to duplicate.

Procedure to Duplicate a Version:

1. Select Version to create a new version in the same Rule.
2. Enter a unique name for the version.
3. (Optional) Enter a brief description for the version.
4. Select the effective start and end dates using the date picker. Alternatively, you can enter them in the space provided.

Caution: The new version's date range must not overlap with any of the existing version date ranges.

5. Click Finish.

Procedure to Duplicate a Rule and Version:

1. Select Rule and Version to create a new rule and a version.
2. Select the folder in which the rule will be stored.
3. Enter a name for the rule.
4. Enter a name for the version.
5. (Optional) Enter a description for the version.
6. Update the effective start and end dates using the date picker. Alternatively, you can type them in the space provided.
7. Click Finish.

Related Topics

Overview of Common Rule Management Tasks, page 2-1

Deleting Rules

You can delete rules that are no longer needed. To delete a rule, you delete all of the versions that are associated with that rule.

Caution: Once deleted, a rule cannot be retrieved.

Restrictions on deleting rules or versions are:

- You cannot delete rules or versions if you have only Read privileges. Only approvers or users with similar or higher system rights can delete rules or versions.
- A rule or version that has been approved can be deleted only by having a delete request approved by the approver.
- You cannot delete rules or versions if their approval is pending. Alternatively, you can delete the approval request and then the rule. However, this works only if you have sufficient privileges.
- You cannot delete versions associated with Locked Rules. A Locked Rule is one that has been already used in the production environment to generate final results.

Procedure:

1. Navigate to the home page, page 2-2 of the rule you want to delete.
2. Search for a rule. For further information, see Searching for Rules, page 2-4.
3. Click Delete corresponding to the rule or the version of the rule that you want to delete.

Related Topics

Overview of Common Rule Management Tasks, page 2-1

Exporting, Importing, and Migrating Rules

Enterprise Performance Foundation provides a framework for exporting and importing rules within a single environment and for migrating and importing rules across multiple

environments. For example, you can develop and test business rules in a testing environment and then migrate those rules to a production environment.

The Export/Import/Migrate features can also handle sub-objects. For example, the Mapping Rule type “Retrieve Statistics” is comprised of a mapping rule and a statistic definition. The statistic definition also includes a Condition. When you migrate or import the “Retrieve Statistics” rule, both the statistic definition and its condition will automatically be included in the process.

The **Export** and **Migrate** buttons, lets you initiate a rule export or migration. Using the Rule Import page you can import and export or migrate rules to the specified user’s folder.

You can export, import, or migrate the following types of rules:

- Dimension and Hierarchy Rules (facilitates moving metadata from one environment to another)
- Condition Rules
- Mapping Rules (available in Oracle Profitability Manager through the Activity-Based Management responsibility)
- Transfer Pricing Rules (available through the Financial Transfer Pricing responsibility)

Prerequisites, Guidelines, and Considerations

In order to import and import or migrate rules successfully, you must first ensure that the following requirements are met and that the systems are set up properly:

- Both the source and target environments should have the same set of languages installed. When you export a rule, only one language will be exported at a time.
- There should be a public database link between the source and target databases when migrating rules between environments. The database link should be registered in Enterprise Performance Foundation (see Working with Registered Database Links, page 10-9 for more information.)
- Only rules with an Approval Status of: **New**, **Approved**, or **Not Approved** can be exported or migrated. Rules with an Approval Status of **Submitted for Approval** or **Submitted for Deletion** cannot be exported or migrated.
- When migrating rules between environments:
 - The username in the source environment must be the same as the username in the target environment.

Note: Potentially, any user in the target environment can import the rule and will have **Read & Write** access to it. However, if the rule was created in the source environment as **Read Only**, then the user who created the rule will be prevented from updating their rule unless that same user performs the import.

If the rule already exists as **Read Only** in the target environment, the import will only be successful if the importing username is the same as the username of the existing target rule.

- You should have folders with the same names in both environments.

Important: Access privileges to these folders should be set to **Read & Write**.

- The global value set combinations in both environments must match.
- The source and target environments should contain the same dimension members, groups, attributes, attribute Versions, and Hierarchies.

Tip: Use the dimension and hierarchy migration tool to migrate metadata from the source environment to the target environment.

- All hierarchies used in a rule should also be present in the target import environment. Importing a migrated rule with a hierarchy must have the same dimension members and hierarchy in place.
- (Oracle Profitability Manager users only) The target import environment's Activity and Cost Object composite dimension definitions must be identical to the definitions in the source migrate environment.

For example, assume that the activity dimension is defined as: **CCC_ORG + Task + User Dimension_01** in the source migrate environment. If the target import environment has Activity defined as: **CCC_ORG + Task**, the import will fail.

- When importing a rule that contains sub-objects in multiple folders, you must have **Read & Write** privileges for all of the folders used in the migrated or exported Rule.

For example, a mapping rule in **Folder1** uses a condition that was created in **Folder2**. **User1** exports or migrates the rule. **User2** must have **Read & Write** access to both folders in order to import the rule

- If importing a block of exported or migrated rules results in errors or warnings, try importing the records one at a time.

The Export, Import, and Migrate Process

The export, import, and migrate process is a simple two-step operation:

1. In the source environment, export or migrate a version of a rule by clicking **Export/Migrate**.
2. In the target environment, import the version using the Rule Import feature

Step 1: Create a new rule or select a version of an existing rule

Note: If Workflow is enabled, select a version with an Approval Status of: **New**, **Approved**, or **Not Approved**.

For example,

1. Create a new condition in the default folder.
2. Filter on the "Company Cost Center Organization" and "Financial Element Dimension" components.
3. Click **Export** to export the rule within a single database or click **Migrate** to migrate the rule between databases.

4. If you clicked **Migrate** in the previous step, select the Target Database

Note: This was enabled by setting up a database link between two environments. See Working with Registered Database Links, page 10-9 for more information.

5. Confirm that you want to export or migrate the condition
6. Monitor the progress of the export or migration

Note: When an Export or Migrate process is run, the system will generate two requests. For export it generates requests for **Export Business Rule** and **iSetup Exporter**. For migrate, it generates requests for **Migrate Business Rule** and **iSetup Exporter**. If the process ends with a warning or an error, then review both request logs.

Step 2: Use Rule Import to select the Exported or Migrated rule for Import

Once the rule has been submitted for import, access the View Request subtab to follow the progress of the import.

If the rule has been imported successfully, subsequent imports of that rule will either overwrite the rule in the folder if it exists for the same effective date range or add itself to the rule when the effective date ranges are different.

An exported rule cannot be imported into a rule where it will cross the effective date ranges of any two existing versions within the rule.

Furthermore, an exported rule cannot be imported into a rule where it will overwrite the effective date range of a rule that has been used in a calculation (a locked rule).

Imports of a migrated rule follow the same process and requirements as those for an exported rule. You should pay special attention to the following requirements:

- The username of the target environment should be the same as the username of the source environment (Highly recommended, though not required; it is especially important in the case of rules with **Read Only** access).
- The Folder name of the Target environment must be the same as the Folder name of the Source environment
- The user must have **Read & Write** access to the folder of the target environment
- The dimensions and dimension members of the target environment must be the same as those in the source environment
- The hierarchies of the target environment must be the same as those in the source environment.

Import, export, or migration of a rule with sub-objects has the following distinctions:

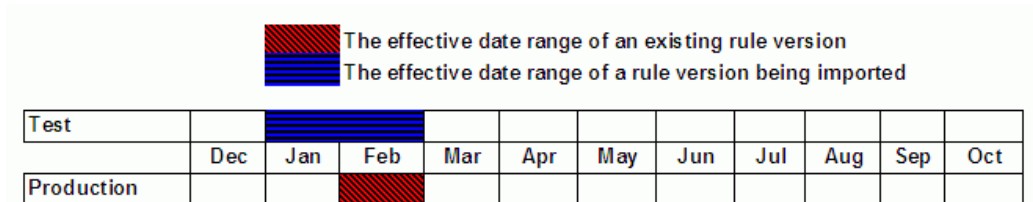
- When importing, the user must have **Read & Write** access to the folder that contains any sub-objects of a rule
- If several rules share the same sub-objects, an Import of the Export/Migrate will only create one copy of the sub-object.
- If several versioned sub-objects fall within the effective date range of the selected parent rule's version, they will be exported or migrated.

- In general, if an error is encountered during export, migration, or import of a parent or sub-object, the export, migration or import will fail for both the parent and all its sub-objects, and the system will write appropriate error messages. This would include the case where no versioned sub-objects fall within the effective date range of the parent version.
- If the sub-object already exists for the same effective date range, the import of the sub-object will update the sub-object and prefix the name with "Import".
- If the sub-object already exists in the target environment, and the Import effective date of the sub-object is different than the original, importing of the sub-object will adjust the effective dates accordingly.

For example, a rule version in the Test environment is Migrated and then Imported in to the Production environment.

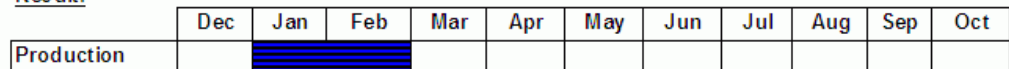
- **Original Rule1 exists in the target Production environment and has one version ("Version1") for 1-Feb-2006 – 28-Feb-2006.**

Import rule version (also named "Version1") with an effective date range 1-Jan-2006 – 28-Feb-2006



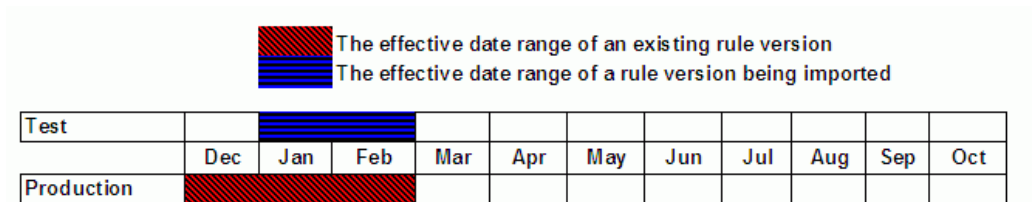
Result: The system will overwrite the target version, creating a version named "(Import) Version1" with the effective date range 1-Jan-2006 – 28-Feb-2006.

Result:



- **Original Rule1 exists in the target environment and has one version ("Version1") for 1-Dec-2005 – 28-Feb-2006.**

Import rule version (also named "Version1") with an effective date range 1-Jan-2006 – 28-Feb-2006.



Result: The system will modify the effective date range of the existing target version, changing it to 1-Dec-2005 – 31-Dec-2005 and will write the imported version (using the name "(Import) Version1"), with the effective date range 1-Jan-2006 – 28-Feb-2006.

Result:

	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct
Production											

- **Original Rule1 exists in the target environment and has one version (“Version1”) for 1-Dec-2005 –31-Mar-2006.**

Import rule version (also named “Version1”) is 1-Jan-2006 – 28-Feb-2006.



The effective date range of an existing rule version
 The effective date range of a rule version being imported

Test											
	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct
Production											

Result: The system will modify the effective date range of the existing target version, changing it to 1-Dec-2005 – 31-Dec-2005, it will write the imported version (using the name “(Import) Version1”) with the imported effective date range of 1-Jan-2006 – 28-Feb-2006, and it will copy the original target version, naming it “(Copy) Version1” and setting the effective date range = 1-Mar-2006 – 31-Mar-2006.

Result:

	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct
Production											

- If the sub-object already exists, the Import of the sub-object cannot be imported where it will cross the effective date ranges of any two existing versions within the rule.
- If the sub-object already exists, the Import of the parent and all sub-objects will fail and log an error message if any import requirements would be violated for the parent or sub-object. For example, if the parent or any sub-object would overwrite the effective date range of a version that has been used in a calculation (a locked version), the system will trigger an error

Dimension and Hierarchy Management

This chapter covers the following topics:

- About Dimension and Hierarchy Management
- Working with Dimension Definitions
- Working with Dimension Attributes
- Working with Dimension Levels
- Working with Members
- Working with Hierarchies
- Viewing Dimension Defaults

About Dimension and Hierarchy Management

To properly represent data in client applications, it is necessary to define metadata (data structures that do not contain actual data values) for the dimensions, hierarchies, levels, attributes, and other structures through the dimension and hierarchy management facility.

For more information, see the following topics:

- Working with Dimension Definitions, page 3-1
- Working with Dimension Attributes, page 3-2
- Working with Dimension Levels, page 3-5
- Working with Members, page 3-7
- Working with Hierarchies, page 3-11
- Viewing Dimension Defaults, page 3-22

Working with Dimension Definitions

You can work with dimension definitions through the Dimension Definition page.

For the dimension currently in view, the Dimension Definition page displays the dimension description, along with the associated defaults for member, hierarchy, and load member. To switch to a different dimension, select the desired dimension in the Switch Dimension box and click the **Go** button.

For further information, see the following topics:

- About dimension default values, page 3-2
- Specifying dimension definitions and default values , page 3-2

About dimension default values

The default values for member, hierarchy, and load member are the values that are used if default values are required when loading the staging tables for data import, as follows:

- The default member is the member used by the application if a user does not specify a member through the user interface.
- The default hierarchy is the hierarchy used by the application as the default hierarchy for the associated dimension.
- The default load member is used to specify the value for a column in the loader table when loading data.

The default values are informational only, and are not used in any Enterprise Performance Foundation user interface pages or concurrent programs. If you are writing your own process to load the staging tables, these defaults may be looked up by your program, if needed.

Specifying dimension definitions and default values

Use the following procedure to modify the definition and default values for a dimension:

1. If the dimension for which you want to modify the definition or default values is not currently displayed on the Dimension Definition page, select the desired dimension in the Switch Dimension box and click the **Go** button.
2. Specify the definition and default values for the dimension as follows:
 - If you want to change the description, modify the existing description or enter a new description in the Description field.
 - If you want to specify a different default member, enter the name of the member in the Default Member field or click the search icon to search for the desired member.
 - If you want to specify a different default hierarchy, enter the name of the hierarchy in the Default Hierarchy field or click the search icon to search for the desired hierarchy.
 - If you want to specify a different default load member, enter the name of the load member in the Default Load Member field or click the search icon to search for the desired load member.
3. When you have finished, click **Apply** to apply the definitions you have specified.

Working with Dimension Attributes

You can work with dimension attributes through the Dimension Attributes page.

The Dimension Attributes page lists the attributes for the dimension currently in view. To switch to a different dimension, select the desired dimension in the Switch Dimension box and click the **Go** button.

To display further information about an attribute, click on the code for the attribute in the Code column.

Note: There are pre-defined, required attributes associated with members for certain dimensions. These attributes cannot be updated or deleted.

For further information, see the following topics:

- Specifying default versions for attributes, page 3-3
- Creating attributes, page 3-3
- Updating attributes, page 3-4
- Deleting attributes, page 3-5

Specifying default versions for attributes

Use the following procedure to specify the default version for an attribute:

1. On the Dimension Attributes page, click on the code for the attribute in the Code column.
2. On the View Dimension Attribute page, select the version that you want to be the default.
3. Click the **Update** button to update the default version for the attribute.

Creating attributes

Use the following procedure to create an attribute:

1. If the dimension for which you want to create an attribute is not currently displayed on the Dimension Attributes page, select the desired dimension in the Switch Dimension box and click the **Go** button.
2. On the Dimension Attributes page, click the **Create** button.
The Create Dimension Attribute page appears.
3. In the Code box, specify a short identifier for the attribute.
4. In the Name box, specify a descriptive name for the attribute.
5. If you want to provide a text description for the attribute, enter it in the Description field.
6. In the Order Type box, select the order type for the attribute. The order types are as follows:
 - **Cardinal** — There is a defined order for the attribute values, but it is not based on alphabetic or numeric order. An example of an attribute using this order type would be shirt size, where the values are small, medium and large.
 - **Nominal** — There is no implied order for the attribute values.
 - **Ordinal** — The ordering of the attribute values is based on the values themselves. An example of an attribute using this order type would be rank, where the values are 1, 2, 3, 4, and 5.
7. In the Data Type box, select the data type for the attribute. The data types are as follows:
 - **Date** — The domain for the attribute is any valid date.

- **Dimension** — The domain for the attribute is any dimension.
If you select the **Dimension** data type, the Dimension field appears, in which you can select a dimension as the source for the domain of the attribute.
 - **Number** — The domain for the attribute is any valid number.
 - **Varchar2** — The domain for the attribute is any alphanumeric string.
8. Select any of the following options that you want to specify:
 - **Allow Multiple Assignment** — The attribute can be assigned to a dimension multiple times. For example, if you want to use a location attribute to specify shipping location and receiving location, you could assign the location attribute twice for the Geography dimension.
 - **Required Attribute** — The attribute is associated with seeded data. Required attributes cannot be created, updated, or deleted by end users.
 - **Allow Multiple Versions** — Allows you to create multiple versions of the attribute. By doing so, you can use a single attribute and still be able to differentiate among versions of that attribute. You must always select the default attribute version.

Note: Other options may be available in specific applications. See the appropriate product documentation for information about such options.
 9. In the Attribute Versions section, select the version that you want to use as the default, then click the **Add a Version** button.
 10. Click the **Apply** button on the Create Dimension Attribute page to create the attribute.

Updating attributes

Use the following procedure to update an attribute:

1. If the dimension for which you want to update an attribute is not currently displayed on the Dimension Attributes page, select the desired dimension in the Switch Dimension box and click the **Go** button.
2. On the Dimension Attributes page, click the **Update** icon for the attribute that you want to update.

The Update Dimension Attribute page appears.

3. Make any desired changes to the settings for the attribute. The attribute settings on the Update Dimension Attribute page are the same as those on the Create Dimension Attribute page. For further information about these settings, see *Creating attributes*, page 3-3.

Note: If the attribute is currently used by any member, you cannot update the data type for the attribute, nor can you specify a different value in the Dimension field (where the data type for the attribute is **Dimension**).

4. Click the **Apply** button to apply the updates to the attribute.

Deleting attributes

You can delete an attribute only if it has not been assigned to a dimension and does not have a value.

Use the following procedure to delete an attribute:

1. If the dimension for which you want to delete an attribute is not currently displayed on the Dimension Attributes page, select the desired dimension in the Switch Dimension box and click the **Go** button.
2. On the Dimension Attributes page, click the **Delete** icon for the attribute that you want to delete, then choose **Yes** in response to the confirmation prompt.

Working with Dimension Levels

You can work with dimension levels through the Dimension Levels page.

The Dimension Levels page lists the levels for the dimension currently in view. To switch to a different dimension, select the desired dimension in the Switch Dimension box and click the **Go** button.

To display further information about a level, click on the name for the level in the Name column.

For further information, see the following topics:

- Reordering levels, page 3-5
- Creating levels, page 3-6
- Updating levels, page 3-6
- Deleting levels, page 3-7

Reordering levels

You can reorder levels only if the levels to be reordered have not been used in a hierarchy.

To reorder levels, do the following:

1. If the dimension for which you want to reorder levels is not currently displayed on the Dimension Levels page, select the desired dimension in the Switch Dimension box and click the **Go** button.
2. On the Dimension Levels page, click the **Reorder Level** button.
The Reorder Dimension Level page appears. This page lists the dimension levels in sequential order.
3. Select a level that you want to place in a different position in the order.
4. Use the arrow buttons to move the selected level to a different position in the order.
5. Repeat the two previous steps for any other levels whose position you want to change.
6. When you have finished reordering the levels, click the **Apply** button to apply the changes and return to the Dimension Levels page.

Creating levels

Use the following procedure to create a level:

1. If the dimension for which you want to create a level is not currently displayed on the Dimension Levels page, select the desired dimension in the Switch Dimension box and click the **Go** button.
2. On the Dimension Levels page, click the **Create Level** button.
The Step 1: Create Dimension Level page appears.
3. Specify the following:
 - A name for the level in the Name field.
 - A text description for the level in the Description field.
 - A short identifier for the level in the Level Display Code field.
4. Select the **Enabled Flag** option if you want the level to be available for selection by users. If you do not select this option, the level will not be available to users.
5. Select any attributes that you want to associate with the level by moving the desired attribute or attributes from the Available Attributes box to the Selected Attributes box.

Note: Required attributes are required for every level.

When creating dimension levels, users are able to assign user-defined attributes to specific levels. These attribute assignments are utilized when a user creates a level-based hierarchy, and restrict the inclusion of attributes on members if those attributes have not been assigned to that level. For example, suppose there is a City level in the Geography dimension and there is a user-defined attribute called Mayor. If a mayor can only be assigned for a city, it is logical to make the Mayor attribute available for dimension members defined at the City level.

6. Click the **Continue** button.
The Step 2: Reorder Dimension Level page appears. This page lists the dimension levels in sequential order.
7. If you want to change the order of the levels, select a level and use the arrow buttons to move the level to a different position in the order. Do this as many times as necessary to achieve the desired order of levels.
8. Click the **Finish** button to create the level and return to the Dimension Levels page.

Updating levels

Use the following procedure to update a level:

1. If the dimension for which you want to update a level is not currently displayed on the Dimension Levels page, select the desired dimension in the Switch Dimension box and click the **Go** button.
2. On the Dimension Levels page, click the **Update** icon for the level that you want to update.

The Update Dimension Level page appears.

3. Make any desired changes to the settings for the level. The settings on the Update Dimension Level page are the same as those on the Create Dimension Level page. For further information about these settings, see *Creating levels*, page 3-6.

Note: You cannot change the setting for the Level Display Code on the Update Dimension Level page.

4. Click the **Apply** button to apply the updates to the level.

Deleting levels

You can delete a level *only* if the level has not been used in a hierarchy. Use the following procedure to delete a level:

1. If the dimension for which you want to delete a level is not currently displayed on the Dimension Levels page, select the desired dimension in the Switch Dimension box and click the **Go** button.
2. On the Dimension Levels page, click the **Delete** icon for the level that you want to delete, then choose **Yes** in response to the confirmation prompt.

Working with Members

You can work with dimension members through the Dimension Members page.

The Dimension Members page lists the members for the dimension currently in view. To switch to a different dimension, select the desired dimension in the Switch Dimension box and click the **Go** button.

To display further information about a member, click on the code for the member in the Code column.

You can update existing members, but you cannot delete them. However, you can disable members; calculations do not write data for disabled members.

Note: There are pre-defined, required attributes associated with members for certain dimensions. These attributes cannot be updated or deleted.

For further information, see the following topics:

- Searching for members, page 3-7
- Creating members, page 3-8
- Updating members, page 3-11

Searching for members

You can search for members on the Dimension Members page. The Dimension Members page provides two types of search facilities, simple and advanced.

Simple search facility

This is the default search facility. To perform a search using the simple search facility, do the following:

1. In the Name box, type the name of the member for which you want to search.

Note: For the Calendar Period dimension, select a calendar in the Calendar box. Optionally, you can also specify a name as search criteria for the Calendar Period dimension.

2. Click the **Go** button to display a list of the members that meet the search criteria.

Advanced search facility

To use this search facility, click the **Advanced Search** button to display the Advanced Search page, then do the following.

1. On the Advanced Search page, specify the desired search parameters.
2. Click the **Go** button to display a list of the members that meet the search criteria.

Creating members

Use the following procedure to create a member:

1. If the dimension for which you want to create a member is not currently displayed on the Dimension Members page, select the desired dimension in the Switch Dimension box and click the **Go** button.

2. On the Dimension Members page, click the **Create Member** button.

The Step 1: Create Dimension Member page appears.

3. Do the following:
 - Specify a short identifier for the member in the Code field.
 - Specify a name for the member in the Name field.
 - If you want to provide a text description for the member, enter it in the Description field.
 - If the member is to be part of a level-based hierarchy, select the level for the member from the list in the Level box.
 - Select the **Enabled Flag** option if you want the member to be included in output. If you do not select this option, the member will not be included in output.
 - If you want to copy the attribute assignment from another member, click the **Search** icon for the Copy Attribute Assignment From box and select the member from which you want to copy the attributes.

4. Click the **Continue** button.

The Step 2: Enter Attributes page appears.

5. Specify values for the attributes. If a search or date picker icon appears next to the value field, you can click the icon to search for a value.

Note that required attributes may appear in the list of attributes. Required attributes can be distinguished from other attributes by the fact that they cannot be deleted.

6. If you want to add additional attributes, select an attribute in the Add box, click the **Go** button, and specify a version and a value for each attribute that you want to add.
7. Click the **Finish** button to create the member and return to the Dimension Members page.

Creating Members through WebADI

Note: For a user intending to use WebADI for creating dimension members and hierarchies, it is mandatory to have access to ‘Data Integration’ folder.

Dimension members can be created through the Dimension Member page.

The Dimension Members page lists the members for the dimension currently in view. To switch to a different dimension, select the desired dimension in the Switch Dimension box and click **Go**.

Use the following procedure to create a dimension member through WebADI:

1. On the dimension members’ page click **Create From Spreadsheet**.

The Excel spreadsheet containing the field names appears.

Note: Macros should be enabled to view the fields in the excel spreadsheet.

2. Save the spreadsheet to your hard drive. This spreadsheet can now be used for uploading data anytime.

WebADI lets you work on the Excel Spreadsheet even without logging on to the application as follows:.

1. Open the spreadsheet and enter data
2. Click **Oracle >Upload**
3. Enter the Login ID and Password.
4. Select the Responsibility and continue

Note: The security option in Excel Spreadsheet should be set to Medium (Tools > Macro > Security > Medium) to enable the users to view the corresponding fields in the Excel Spreadsheet

3. The header section in the spreadsheet contains the Dimension, Dimension Code and Ledger fields.

Field	Functionality
Dimension	This gives the name of the dimension for which the member needs to be created
Dimension Code	This gives the dimension name used for internal references
Ledger	Users need to select a ledger since dimension loaders are used to push members into the database.

- The preference selection for the ledger is not applicable in the spreadsheet, even though you launch the spreadsheet from the UI, where that ledger was specified.

- In case the dimension selected is Calendar period, Ledger would be replaced by Calendar.

4. The table contains the following details:

Column	Details
Upload	Flag icon is displayed when the user enters the Member Name
Member Name	Users need to specify a member name here
Member Code	Users need to specify a short identifier for the member
Description	Text Description for the member is provided here

If the member is to be part of a level-based hierarchy, select the level for the member from the list in the Level column.

Note: Selection of Level is mandatory in case of Calendar period Dimension member creation

5. Specify values for attributes. All the attributes, which have an asterisk against its name, are mandatory. The remaining are optional.

Note: The non-required attributes not associated with selected level will be ignored during the upload.

6. Place the cursor on the field and double click to view the List Of Values available for the user. (Alternatively, the List of Values would be displayed by clicking on Oracle > List of Values in the menu)

Note: User defined attributes can be made available to the users by adding them through the dimension attribute page and running a concurrent program (Run the Concurrent program "Maintain Web ADI Metadata for Dimension Member Integration" to load the user defined attributes.)

7. Click on Oracle > Upload in the menu.

8. You have the option to select 'All Rows' for Upload, which means that all the rows available in the Excel spreadsheet would be uploaded onto the application

Note: Users may create as many members as desired by unprotecting the sheet and inserting the rows)

9. Selecting the 'Flagged Rows' radio button enables loading only those rows which contain a Flag against it within the excel spreadsheet.

10. View the status of the Concurrent Request by clicking on Oracle > Monitor in the menu

11. Return to the Dimension members page and check for the new member created

12. The Messages column gives the corresponding details of the success or failure of member upload.

In case the program fails, users need to check the log files for more details.

- WebADI allows users to save the excel spreadsheet onto the desktop for future usage (creation and updating).
- WebADI allows creation of multiple members through the same spreadsheet
- When we try to load a member with Name and Code of an existing member, the new member overwrites the existing member.

Updating members

Use the following procedure to update a member:

1. If the dimension for which you want to update a member is not currently displayed on the Dimension Members page, select the desired dimension in the Switch Dimension box and click the **Go** button.
2. On the Dimension Members page, click the **Update** icon for the member that you want to update.

The Update Dimension Member page appears.

3. Make any changes to the settings for the member. The settings on the Update Dimension Member page are the same as those on the Create Dimension Member page. However, there are two restrictions that apply to updating members:
 - You cannot change the setting for Code on the Update Dimension Member page.
 - If the member has been used in a level-based hierarchy, you cannot change the setting for the level on the Update Dimension Member page.

For further information about the settings on the Create Dimension Member page, see Creating members, page 3-8.

4. Click the **Apply** button to apply the updates to the member.

Working with Hierarchies

You can work with hierarchies through the Hierarchies page.

The Hierarchies page lists the hierarchies for the dimension currently in view. To switch to a different dimension, select the desired dimension in the Switch Dimension box and click the **Go** button.

To display the version or versions for a hierarchy, click the **Expand** icon immediately preceding the hierarchy name. For each version, the Hierarchies page shows the status, start date, and end date for that version.

To display further information about a hierarchy or version, click on the name of the hierarchy or version in the Name column.

For further information, see the following topics:

- Searching for hierarchies, page 3-12
- Creating hierarchies, page 3-12

- Updating hierarchy definitions, page 3-15
- Updating hierarchy version definitions, page 3-16
- Working with hierarchy version details, page 3-16
- Duplicating hierarchies and hierarchy versions, page 3-18
- Deleting hierarchies and hierarchy versions, page 3-19

Searching for hierarchies

You can search for hierarchies on the Hierarchies page. The Hierarchies page provides two types of search facilities, simple and advanced. To perform a search when you do not know the name of the value for which you want to search, specify a percent (%) sign as the search value.

Simple search facility

This is the default search facility. To perform a search using the simple search facility, do the following:

1. In the Folder box, select the folder in which you want to search. If you do not select a folder, all folders to which you have access will be searched.
2. In the Hierarchy Name box, type the name of the hierarchy for which you want to search.
3. Click the **Go** button to display a list of the hierarchies that meet the search criteria.

Advanced search facility

To use this search facility, click the **Advanced Search** button to display the Advanced Search page, then do the following.

1. On the Advanced Search page, use the Folder box to select the folder in which you want to search. If you do not select a folder, all folders to which you have access will be searched.
2. Specify the desired search parameters.
3. Click the **Go** button to display a list of the hierarchies that meet the search criteria.

Creating hierarchies

You can create hierarchies through the Create Hierarchy Wizard. Click the **Create Hierarchy** button on the Hierarchies page to open the Create Hierarchy Wizard.

The Create Hierarchy Wizard consists of the following steps:

- Step 1: Create definition, page 3-13
- Step 2: Select levels, page 3-14
- Step 3: Select value sets, page 3-14
- Step 4: Create version, page 3-14
- Step 5: Select root nodes, page 3-14
- Step 6: Review definition, page 3-15

Step 1: Create definition

The first step in the Create Hierarchy Wizard is the Create Definition step, which allows you to create the definition for the hierarchy. In this step, do the following:

1. In the Folder box, select the folder in which you want to create the hierarchy.
2. In the Name box, specify the name for the hierarchy.
3. In the Description box, specify a text description for the hierarchy (optional).
4. In the Level Use Code box, select one of the following:
 - **No Levels** — Specifies that the sequence of previously-defined levels is not enforced for this hierarchy. This means that there are no level-related restrictions on family relationships defined by users.
 - **Sequence Enforced** — Specifies that the sequence of previously-defined levels is enforced for the hierarchy, and that no levels can be skipped. For example, if the previously-defined hierarchy is State > County > City, a user defining a family relationship must also use State > County > City.
 - **Sequence Enforced - Skip Level** — Specifies that the order of previously-defined levels is enforced for the hierarchy, but levels can be skipped. For example, if the previously-defined hierarchy is State > County > City, a user could not define a relationship of City > State, but a family relationship of State > City would be allowed.
5. In the Hierarchy Type box, select one of the following:
 - **Open** — This is the standard hierarchy type. There are no special functions or restrictions associated with it.
 - **Reconciliation** — Reconciliation hierarchies are used in conjunction with members for which the RECON_LEAF_NODE_FLAG is set to Y. These hierarchies prohibit children from being added below nodes for which the member is defined as a leaf. Such hierarchy types are typically used to ensure that all leaf-level members are included in a specific hierarchy.
6. In the Hierarchy Usage Code box, select a setting to specify how the hierarchy is to be used. These values for Hierarchy Usage Code are simply informational properties for the hierarchy that are used to indicate its purpose.
7. If the hierarchy is for the Calendar Period dimension, select a calendar for the hierarchy in the Calendar box. The Calendar box is not displayed for other dimensions.
8. Select any of the following options that you want to specify:
 - **Read Only or Read & Write** — Select one of these options to specify the user access rights for the hierarchy.
 - **Allow Multiple Tops** — Select this option if you want to allow more than one top-level parent value in the hierarchy.
 - **Allow Multiple Value Sets** — Select this option for hierarchies created for the Financial Consolidation Hub.
 - **Save in standard and flattened format** — Specifies how the hierarchy is saved. Specifying this option can improve performance, although more data storage is required.
9. Click the **Next** button.

Step 2: Select levels

The second step in the Create Hierarchy Wizard is the Select Levels step, which allows you to select the levels that you want to include in the hierarchy.

Note: If you selected **No Levels** for Level Use Code in the Create Definition step, the Create Hierarchy Wizard omits the Select Levels step.

In the Select Levels step, do the following:

1. Select the levels that you want to include in the hierarchy.
2. Click the **Next** button.

Step 3: Select value sets

The third step in the Create Hierarchy Wizard is the Select Value Sets step, which allows you to select the value sets that you want to use for the hierarchy.

Note: If you selected the **Allow Multiple Value Sets** option in the Create Definition step, the Create Hierarchy Wizard omits the Select Value Sets step.

In the Select Value Sets step, do the following:

1. Select the value sets that you want to use for the hierarchy.
2. Click the **Next** button.

Step 4: Create version

The fourth step in the Create Hierarchy Wizard is the Create Version step, which allows you to define the version of the hierarchy. In this step, do the following:

1. In the Version box, specify the name for the version.
2. In the Description box, specify a text description for the version (optional).
3. In the Start Date box, click the date picker icon to select the start date for the version, or manually enter the start date. If the profile option FEM:Default Effective Start Date has been entered (through E-Business Suite user interfaces), that value is used in the Start Date box.
4. In the End Date box, click the date picker icon to select the end date for the version, or manually enter the end date. If the profile option FEM:Default Effective End Date has been entered (through e-Business Suite user interfaces), that value is used in the End Date box.
5. Click the **Next** button.

Step 5: Select root nodes

The fifth step in the Create Hierarchy Wizard is the Select Root Nodes step, which allows you to select one or more root nodes (top-level members) for the hierarchy, as follows:

- If you selected the **Allow Multiple Tops** option in the Create Definition step, you may specify more than one root node.
- If you did not select the **Allow Multiple Tops** option in the Create Definition step, you must specify only one root node.

In this step, do the following:

1. Do one of the following:
 - In the Available Members box, select the member or members that you want to define as root nodes, depending on whether you selected the **Allow Multiple Tops** option in the Create Definition step.

To filter out unwanted members from a long list of member names, thereby making it easier to find a particular member, specify the name of the member that you want to find in the Filter box, then click the **Go** button.

When you have selected the member or members in the Available box, click **Move** to move the selected member or members to the Selected Members box.
 - Click **Move All** to select all of the members listed in the Available Members box as root nodes.
2. If you want to reorder the members in the Selected Members box, select one or more members and use the move icons in the Selected Members box to move the members in the sequence.
3. Click the **Next** button.

Step 6: Review definition

The last step in the Create Hierarchy Wizard is the Review Definition step, which allows you to review the definition for the hierarchy before completing the hierarchy creation process. In this step, do the following:

1. Review the hierarchy definition. If you want to make any modifications to the definition, use the **Back** buttons to return to the steps that you want to change.
2. When you are satisfied that the hierarchy definition is correct, click the **Finish** button to create the hierarchy.

Updating hierarchy definitions

Use the following procedure to update a hierarchy definition:

1. If the dimension for which you want to update a hierarchy definition is not currently displayed on the Hierarchies page, select the desired dimension in the Switch Dimension box and click the **Go** button.
2. On the Hierarchies page, click the **Update** icon for the hierarchy for which you want to update the definition.
3. On the Update Hierarchy Definition page, make any desired changes to the name and the description for the hierarchy definition. In certain cases, you can also make other changes to the hierarchy definition, as follows:
 - If you are the user who originally created the hierarchy definition, you have the option to change the user access rights from **Read Only** to **Read & Write** or from **Read & Write** to **Read Only**. Only the user who originally created the hierarchy can change the access rights.
 - If the **Save in standard and flattened format** option has not previously been selected, you can select it now if you want to save the hierarchy in standard and flattened format. Specifying this option can improve performance, although more data storage is required. If you select the option here, a concurrent request

is generated to launch a process that saves the hierarchy in this format; while the process is running, all versions of the hierarchy display a status of Pending.

4. Click the **Apply** button to save the changes.

Updating hierarchy version definitions

Use the following procedure to update a hierarchy version definition:

1. If the dimension for which you want to update a hierarchy version is not currently displayed on the Hierarchies page, select the desired dimension in the Switch Dimension box and click the **Go** button.
2. On the Hierarchies page, display the versions for the hierarchy for which you want to update a version by clicking the **Expand** icon for the hierarchy.
3. Click the **Update** icon for the version that you want to update.
4. On the Hierarchy Version Definition page, make any desired changes to the description, start date, and end date for the version (you cannot make any other changes to the definition).
5. Click the **Continue** button.
6. On the Select Root Nodes page, make any desired changes to the selected root node members. Note that you can add root nodes only to a multi-top hierarchy, and you cannot remove root nodes from any hierarchy type.
7. Click the **Finish** button.

Working with hierarchy version details

The Version Details page allows users to perform the various tasks related to hierarchy versions.

To display the Version Details page for the desired hierarchy version, do the following:

1. If the dimension for which you want to update version details for a hierarchy is not currently displayed on the Hierarchies page, select the desired dimension in the Switch Dimension box and click the **Go** button.
2. On the Hierarchies page, display the versions for the hierarchy for which you want to update version details by clicking the **Expand** icon for the hierarchy.
3. Click the **Details** icon for the version that you want to update to display the Version Details page for the hierarchy version.

On the Version Details page, you can use the **Expand** icons to display any of the child nodes. You can also use the **Focus** icons to focus on a particular node (and its children).

For further information, see the following topics:

- Searching for nodes, page 3-17
- Printing version details, page 3-17
- Adding nodes, page 3-17
- Reordering children, page 3-17
- Moving nodes, page 3-18
- Deleting nodes, page 3-18

Searching for nodes

To search for nodes for which you want to display version details, do the following:

1. On the Version Details page, click the **Search Nodes** button.
2. On the Search Nodes page, do one of the following:
 - Use the standard search facility to search for the name of the node or nodes for which you want to display version details.
 - Click the **Advanced Search** button to use the advanced search facility, which allows you to search for nodes based on additional search criteria.
3. Click the **View Children** icon for the node for which you want to display version details.

Printing version details

To produce a version of the Version Details page that is formatted for printing, do the following:

1. On the Version Details page, click the **Printable Page** button.
2. Use your browser's print function to print the version details.
3. Use your browser's back button to return to the Version Details page.

Adding nodes

To add child nodes to a node, do the following:

1. Click the **Add** icon for the node to which you want to add child nodes.
2. On the Add Nodes page, do one of the following:
 - Use the standard search facility to search for the name of the node or nodes that you want to add.
 - Click the **Advanced Search** button to use the advanced search facility, which allows you to search for nodes based on additional search criteria.
3. Select the node or nodes that you want to add.
4. Click the **Apply** button to add the selected nodes.

Reordering children

To reorder child nodes for a node, do the following:

1. Click the **Reorder** icon for the node for which you want to reorder the child nodes.
2. On the Reorder Children page, do one of the following:
 - Use the **Move** icons in the standard reorder facility to specify the new order for the child nodes. You can change the order in which the nodes are listed by selecting **Ascending** or **Descending** in the Order By box and clicking the **Go** button.
 - Click **Advanced Reorder** button to use the advanced reorder facility. This allows you to manually enter sequence numbers, which may be the easiest reordering method when there are a large number of nodes to be reordered.

To use the advanced reorder facility, specify a member in the Member box and click the **Go** button to search for the member or group of members, enter the desired sequence numbers, and click the **Apply** button.

3. Click the **Apply** button to reorder the child nodes.

Moving nodes

To move a node, do the following:

1. Click the **Move** icon for the node for the node that you want to move.
The Move Nodes page appears.
2. Search for the target node (the node to which you want to move the node for which you clicked the **Move** icon in the previous step) by doing one of the following:
 - Use the standard search facility to search for the name of the target node.
 - Click the **Advanced Search** button to use the advanced search facility, which allows you to search for a target node based on various search criteria.
3. Click the **Go** button to display a list of potential target nodes.
4. Select the desired target node.
5. Click the **Apply** button to move the selected node.

Deleting nodes

To delete a node, click the **Remove** icon for the node that you want to delete, then choose **Yes** in response to the confirmation prompt.

Duplicating hierarchies and hierarchy versions

You can create duplicates of either of the following:

- A hierarchy and version
- A version only

Creating and then modifying a duplicate provides a simple method for creating an entirely new hierarchy and version, or a new version of an existing hierarchy, based on an existing hierarchy and version.

For further information, see the following topics:

- Creating a duplicate of a hierarchy and version, page 3-18
- Creating a duplicate of a version, page 3-19

Creating a duplicate of a hierarchy and version

To make a duplicate of a hierarchy and version, do the following:

1. On the Hierarchies page, click the **Duplicate** icon for the version from which you want to create a duplicate.
2. On the Step 1 Duplicate Options page, select the **Hierarchy and Version** option.
3. Click the **Continue** button to display the Step 2 Duplicate New Hierarchy and Version page.

4. In the Folder box, select the folder in which you want to create the duplicate hierarchy and version.
5. In the Hierarchy Name box, specify the name for the duplicate hierarchy.
6. In the Hierarchy Description box, specify a text description for the duplicate Hierarchy (optional).
7. In the Version Name box, specify the name for the duplicate version.
8. In the Version Description box, specify a text description for the duplicate version (optional).
9. In the Effective Start Date and Effective End Date boxes, click the date picker icons to select the start and end dates for the duplicate version, or manually enter the start date and end date in the format dd-MMM-yy.
10. Click the **Apply** button to create the duplicate hierarchy and version.

Creating a duplicate of a version

To make a duplicate of a version only, do the following:

1. On the Hierarchies page, click the **Duplicate** icon for the version from which you want to create a duplicate.
2. On the Step 1 Duplicate Options page, select the **Version** option.
3. Click the **Continue** button to display the Step 2 Duplicate Version page.
4. In the Version Name box, specify the name for the duplicate version.
5. In the Start Date and End Date boxes, click the date picker icons to select the start and end dates for the duplicate version, or manually enter the start date and end date.

Note: The start and end dates that you specify must not overlap with those of existing versions of the same hierarchy.

6. In the Description box, specify a text description for the duplicate version (optional).
7. Click the **Apply** button to create the duplicate version.

Deleting hierarchies and hierarchy versions

You can delete a hierarchy or hierarchy version *only* if the access rights for the hierarchy are **Read & Write**. You cannot delete a hierarchy for which the access rights are **Read Only**.

Use the following procedure to delete a hierarchy or hierarchy version:

1. If the dimension for which you want to delete a hierarchy or hierarchy version is not currently displayed on the Hierarchies page, select the desired dimension in the Switch Dimension box and click the **Go** button.
2. On the Hierarchies page, click the **Delete** icon for the hierarchy or hierarchy version that you want to delete, then choose **Yes** in response to the confirmation prompt.

When you delete a hierarchy or version, a concurrent process is launched to perform the operation. The hierarchy or version remains in the list of hierarchies until the deletion process is complete.

Note: You cannot use the above procedure to delete a hierarchy that has been transferred from Oracle General Ledger through a Hierarchy Rule. For information about deleting such a hierarchy, see *Deleting Hierarchies*, page 12-27.

Creating Hierarchies through WebADI

Note: The WEB ADI integration with the DHM is not meant for high volumes of dimensions and hierarchies. In the situations that require a long list of dimensions members, or very large hierarchies, please utilize the appropriate interface tables and loader programs.

The Hierarchies page lists the hierarchies for the dimension currently in view. To switch to a different dimension, select the desired dimension in the Switch Dimension box and click **Go**.

Use the following procedure to create a Hierarchy through WebADI:

1. On the Hierarchies' page click **Create From Spreadsheet**.

The Excel spreadsheet containing the field names appears.

Note: Macros should be enabled to view the fields in the excel spreadsheet.

2. Save the spreadsheet to your hard drive. This spreadsheet can now be used for uploading data anytime.

WebADI lets you work on the Excel Spreadsheet even without logging on to the application as follows:

1. Open the spreadsheet and enter data
2. Click **Oracle >Upload**
3. Enter the Login ID and Password.
4. Select the Responsibility and continue

Note: The security option in Excel Spreadsheet should be set to Medium (Tools > Macro > Security > Medium) to enable the users to view the corresponding fields in the Excel Spreadsheet

3. The Header section in the spreadsheet contains the following details:

Field	Details
Folder	User selects the folder within which hierarchy may be created
Ledger	Users need to select a ledger since dimension loaders are used to push members into the database.
Dimension	Gives the name of the dimension for which the hierarchy needs to be created
Version	Users need to specify the name for the version
Start Date	Manually enter the start date
End Date	Manually enter the end date
Allow Multiple Tops	Select the value as 'Yes' for this field if you want to allow more than one top-level parent value in the hierarchy

Note: The Ledger would be replaced with Calendar for the Calendar Period Dimension. If the hierarchy is for the Calendar Period Dimension, select a calendar for the hierarchy in the Calendar box. The Calendar is not displayed for other dimensions.

4. In the Level Use Code box, select one of the following:
 - **No Levels** — This means that there are no level-related restrictions on family relationships defined by users.
 - **Sequence Enforced** — Treats the hierarchy created with the specified levels as a Sequence Enforced Hierarchy
 - **Sequence Enforced - Skip Level**— Treats the hierarchy created with the specified levels as a Sequence Enforced - Skip Level Hierarchy.

Note: WebADI Hierarchy Integrator does not provide any validation wrt hierarchies that are Sequence Enforced and Sequence Enforced Skip Level. It provides option to the users to create hierarchies of all types through WebADI which may be further updated / modified through DHM. However, the EPF Loader performs this validation during upload.

5. The Upload column displays Flag icon when the user selects the Parent member for the Hierarchy.
6. Specify the Parent Member by selecting from the List of Values / manually entering them.

Note: The EPF Loader programs nominates only such members as the root nodes for any hierarchy for which the Parent Member=Child Member.

Note: Therefore the users need to specify the above for all the root nodes that they intend to create for a hierarchy.

7. Specify the Child Member by either selecting them from the list of Values or entering them
8. Click **Oracle > Upload** in the menu.
9. You have the option to select **All Rows' for Upload**, which means that all the rows available in the Excel spreadsheet would be uploaded onto the application
10. Selecting the **Flagged Rows'** radio button enables loading only those rows which contain a Flag against it within the excel spreadsheet.
11. View the status of the Concurrent Request by clicking **Oracle > Monitor** in the menu
12. The Messages column gives the details of the success or failure of Hierarchy upload. In case the program fails, users need to check the log files for more details.
 - WebADI allows users to save the excel spreadsheet onto the desktop for future usage (creation and updating).
 - WebADI allows creation of multiple level hierarchies through the same spreadsheet in a single operation
 - Users should unprotect the sheet for entering more rows through Tools > Protection > Unprotect Sheet
 - Return to the Hierarchies page and check for the new hierarchy created

Viewing Dimension Defaults

Through the Load Member Defaults page, you can view the default values assigned to dimensions for ledger data loaded into tables.

Use the following procedure to display default values for a table on the Load Member Defaults page:

1. In the Table box, specify the display name of the table for which you want to display default values.
2. In the Ledger box, select the desired ledger.
3. Click the **Go** button to display the default values for the specified table and ledger.

Using Conditions to Select Data

This chapter covers the following topics:

- About Conditions
- Working with conditions
- Creating Conditions
- Working with Dimension Components
- Working with Data Components

About Conditions

A condition is an object that is used to select data for use as input to a business rule.

A condition must include either or both of the following to define the data selection:

- One or more dimension components
- One or more data components

Conditions use versions to provide different selection characteristics for different sets of start and end dates. Each condition has at least one version, and there can be multiple versions for each condition.

The Conditions page is the starting point from which you can view, create, update, and delete conditions and versions.

For more information, see the following topics:

- Working with Conditions, page 4-1
- Creating Conditions, page 4-4
- Working with Dimension Components, page 4-4
- Working with Data Components, page 4-7

Working with conditions

Through the Conditions page, you can view, duplicate, update, and delete existing conditions and versions.

For more information, see the following topics:

- Viewing conditions and versions, page 4-2

- Updating conditions, page 4-2
- Updating condition versions, page 4-2
- Duplicating versions to create new conditions and versions, page 4-3
- Duplicating versions to create new versions of existing conditions, page 4-3
- Deleting conditions and versions, page 4-4

Viewing conditions and versions

Use the following procedure to search for and display existing conditions and versions on the Conditions page:

1. In the Folder box, select the folder in which you want to search for conditions and versions.

Note: Hierarchies and other components that have previously been created will be available *only* if they have been created in the folder that you select; they will not be available if they were created in a different folder.

2. If you want to refine the search criteria by name, specify all or part of a condition name.
3. If you want to refine the search criteria by date, use the date picker to specify a date in the Effective Date box.
4. Click the **Go** button to display a list of the conditions and versions that meet the search criteria.

You can click on the name of a condition or version in the list to display information about the condition or version.

Updating conditions

You can update a condition by changing the name and description. Use the following procedure to update a condition:

1. On the Conditions page, display the condition that you want to update.
2. Click the **Update** icon for the condition.

An update page for the condition appears.

3. Make any desired changes to the name and description.
4. Click the **Apply** button.

Updating condition versions

You can update a condition version by making changes to the name, the description, and the start and end dates, as well as to any dimension components and data components. Note that you cannot update a version that is process locked (that is, a version used in a calculation to produce data) except to update the effective dates. Use the following procedure to update a version:

1. On the Conditions page, display the version that you want to update
2. Click the **Update** icon for the version.

3. On the on the Condition Definition page, do any of the following, as desired:
 - Make any desired changes to the name, start date, end date, and description.
 - To make changes to dimension components, click the **Dimension Components** button. For further information, see Working with Dimension Components, page 4-4.
 - To make changes to data components, click the **Data Components** button. For further information, see Working with Data Components, page 4-7.
 - If you want to view the SQL that the condition represents, click the **Display SQL** button to submit a concurrent process to display the SQL text. You can then view the output of the concurrent process to see the SQL generated by the condition.
4. Click the **Apply** button

Duplicating versions to create new conditions and versions

You can duplicate a condition version to create a new condition rule, along with a version for the new condition, as follows:

1. On the Conditions page, display the version that you want to duplicate.
2. Click the **Duplicate** icon for the version.
3. On the Step 1: Duplicate Options page, select **Rule and Version**.
4. On the Step 2: Duplicate Rule and Version page, do the following:
 1. Select the folder in which you want to create the new condition and version.
 2. Specify the new names for the condition and version.
 3. Use the date pickers to specify the effective start and end dates for the new version.
 4. If you want to provide a text description for the version, type the description in the Description box.
 5. Click the **Finish** button.

The new condition and version appear on the Conditions page.

Duplicating versions to create new versions of existing conditions

You can duplicate a condition version to create a new version of an existing condition, as follows:

1. On the Conditions page, display the version that you want to duplicate.
2. Click the **Duplicate** icon for the version.
3. On the Step 1: Duplicate Options page, select **Version**.
4. On the Step 2: Duplicate Version page, do the following:
 1. Specify the new version name.
 2. Use the date pickers to specify the effective start and end dates for the new version.
 3. If you want to provide a text description for the new version, type the description in the Description box.

4. Click the **Finish** button to create the new version.

The new version appears on the Conditions page.

Deleting conditions and versions

You can explicitly delete a condition version. To delete a condition, you must delete all of the versions for that condition. Note that you cannot delete a condition if the condition is specified as part of a rule definition.

Use the following procedure to delete a condition version:

1. On the Conditions page, click the **Delete** icon for the version that you want to delete.
2. Click **Yes** in response to the prompt to delete the version.

Creating Conditions

Use the following procedure to create a condition:

1. On the Conditions page, click the **Create** button.
2. On the Step 1: Condition Rule Details page, do the following:
 1. In the Folder box, select the folder in which you want to save the condition.
 2. In the Condition Name box, specify the name for the condition.
 3. If you want to provide a text description for the condition, type the description in the Description box.
 4. Specify the access rights for the condition. If you want all users other than yourself to have read-only access rights, so that they can view the condition but cannot update or delete it, select **Read**. If you want all other users to have read and write access rights, so that they are able to view, update, and delete the condition, select **Read/Write**.
 5. Click the **Continue** button to display the Step 2: Condition Definition page.
3. On the Step 2: Condition Definition page, do the following:
 1. In the Version Details section specify the name, start date, end date, and (optionally) a text description for this version of the condition.
 2. If you want to specify a dimension component, click the **Dimension Components** button.

For further information, see Working with Dimension Components, page 4-4.
 3. If you want to specify a data component, click the **Data Components** button.

For further information see Working with Data Components, page 4-7.
 4. Click **Finish** to create the condition or version.

Working with Dimension Components

A dimension component selects data for a condition by specifying dimension members. You can specify one or more dimension components for a condition.

A dimension component can specify dimension members in one of three ways:

- A dimension component can directly specify a particular member of a dimension. For example, a dimension component could specify that you want to select data for the Building and Improvements member in the Line Item dimension.
- Through a component definition, a dimension component can specify members based on attribute values. For example, a dimension component could specify all members of the Line Item dimension where the Extended Account Type is Expense.
- Through a component definition, a dimension component can specify based on hierarchical relationships. For example, a dimension component could specify all descendants of the node Recoveries - General Allowance in the Line Item dimension.

You specify the dimension components that you want to use for a condition through the Dimension Component Definition page. To access the Dimension Component Definition page, click the **Dimension Components** button on the Step 2: Condition Definition page.

For more information, see the following topics:

- Specifying dimension components, page 4-5
- Creating component definitions that use attribute values, page 4-5
- Creating component definitions that use hierarchical relationships, page 4-6
- Updating dimension components, page 4-7
- Removing dimension components, page 4-7

Specifying dimension components

Use the following procedure to specify a dimension component:

1. On the Dimension Component Definition page, click the **Add Dimension Component** button.

A new row appears on the Dimension Component Definition page.

2. In the Dimension Column field in the new row, select the dimension with which the dimension component is associated.
3. Do one of the following:
 - Select a particular dimension member in the Value field.
 - Select an existing component definition in the Component Definition field.
 - Create a new component definition that you can then select in the Component Definition field. For further information, see Creating component definitions that use attribute values, page 4-5 and Creating component definitions that use hierarchical relationships, page 4-6.

You can specify additional dimension components by following the above procedure for each dimension component that you want to specify.

When you have completed your work on the Dimension Component Definition page, click the **Apply** button to define the dimension components that you have specified.

Creating component definitions that use attribute values

You can create a component definition that uses attribute values to specify members for a given dimension, as follows:

1. On the Dimension Component Definition page, click the **Create Definition** icon in the row for the desired dimension component. Note that the component definition that you create will be for the dimension specified by the Dimension field in the row in which you click **Create Definition**.

The Step 1: Component Definition Type page appears.

2. Specify a name for the component definition in the Component Name field.
3. If you want to provide a text description for the component definition, type the description in the Description box.
4. Under Component Type, select **Attribute**.
5. Click the **Continue** button.

The Step 2: Dimension Attribute page appears.

6. Specify an attribute value for one or more of the listed attributes.
7. Click the **Apply** button to create the component definition.

Creating component definitions that use hierarchical relationships

You can create a component definition that uses hierarchical relationships to specify members for a given dimension, as follows:

1. On the Dimension Component Definition page, click the **Create Definition** icon in the row for the desired dimension component. Note that the component definition that you create will be for the dimension specified by the Dimension field in the row in which you click **Create Definition**.

The Step 1: Component Definition Type page appears.

2. Specify a name for the component definition in the Component Name field.
3. If you want to provide a text description for the component definition, type the description in the Description box.
4. Under Component Type, select **Hierarchy**.
5. Under Hierarchy Version Details, select the hierarchy and version that you want to use.
6. Click the **Continue** button.

The Step 2: Relation page appears.

7. Select one or more nodes that you want to use for the component definition and specify the hierarchical relation for each selected node in the Relation field. The relation types that you can specify are as follows:
 - Descendants Of — All descendants of the selected node, not including the selected node itself
 - Last Descendants Of — Only the descendants of the selected node that are at the lowest level of the hierarchy, not including the selected node itself
 - Node Only — Only the selected node
 - Node and Descendants Of — The selected node and all of its descendants
8. Click the **Apply** button on the Step 2: Relation page to create the component definition.

Updating dimension components

You can update dimension components that use component definitions to specify members. It is not possible to update dimension components that specify a particular dimension member in the Value field.

Use the following procedure to update a dimension component:

1. On the Dimension Component Definition page, click the **Update Definition** icon for the dimension component that you want to update.
2. Do one of the following, depending on whether the component definition for the dimension component uses attribute values or hierarchical relationships:
 - If the component definition uses attribute values, make any desired changes to the specified attribute values on the Step 2: Dimension Attributes page, then click the **Apply** button.
 - If the component definition uses hierarchical relationships, make any desired changes to the specified relationships on the Step 2: Relationships page, then click the **Apply** button.

Removing dimension components

Use the following procedure to remove a dimension component:

1. On the Dimension Component Definition page, click the **Remove** icon for the dimension component that you want to remove.
2. Click **Yes** in response to the prompt to remove the dimension component.

Working with Data Components

A data component selects data for a condition on the basis of values such as dates, numeric values, and so forth. You can specify one or more data components for a condition.

A data component can specify values by defining a relationship between one or more columns in a table (where each column represents a dimension) in a table and one of the following:

- The values in another column
- A range of values
- Specific values

You specify the data components that you want to use for a condition through the Data Component Definition page. To access the Data Component Definition page, click the **Data Components** button on the Step 2: Condition Definition page.

For more information, see the following topics:

- Specifying data components, page 4-8
- Creating data components, page 4-8
- Updating data components, page 4-9
- Removing data components, page 4-9

Specifying data components

Use the following procedure to specify a data component:

1. On the Data Component Definition page, click the **Add Data Component** button.

A new row appears on the Data Component Definition page.

2. In the Data Component field in the new row, select a data component.

You can specify additional data components by following the above procedure for each data component that you want to specify.

When you have completed your work on the Data Component Definition page, click the **Apply** button to define the data components that you have specified.

Creating data components

Use the following procedure to create a data component:

1. On the Data Component Definition page, click the **Create Data Component** button.

2. On the Step 1: Table Column Selection page, do the following:

1. Specify the name for the data component in the Component Name box.

2. If you want to provide a text description for the data component, type the description in the Description box.

3. Use the Search section to specify the tables listed in the Results section.

If you want to narrow the search by name, specify one or more characters of the table name in the TableName field, followed by percent sign (%), then click the **Go** button. You can also choose a particular table from the list in the Table List box. If you do not use either the TableName box or the Table List box to search for a table, the Results section displays all available tables.

4. In the Results section, select one or more tables.

For each table that you select, select one or more columns. To display columns for a table, expand the display for the table by clicking the icon directly to the left of the table name.

5. Click the **Continue** button.

3. On the Step 2: Column Steps Definition page, click the **Create Step** icon for a table (as listed in the Step Column field) to display the Step 1: Step Type page.

4. On the Step 1: Step Type page, select one of the following, depending on the type of relationship that you want to define:

- **Another Column**
- **Range Values**
- **Specific Values**

5. Click the **Continue** button.

6. Depending on the selection you made on the Step 1: Step Type page, do one of the following:

- If you selected **Another Column**, specify a column, an operator, and a value on the Step 2: Another Column page, then click the **Apply** button.

- If you selected **Range Values**, specify a column, an operator, and two values on the Step 2: Range Values page, then click the **Apply** button.
 - If you selected **Specific Values**, specify a column, an operator, and a value on the Step 2: Specific Values page, then click the **Apply** button.
7. Repeat Steps 3 through 6 for each table listed in the Step Column field on the Step 2: Column Steps Definition page. You must create at least one step for each table listed in the Step Column field.
 8. When you have created one or more steps for each table listed in the Step Column field, click the **Finish** button on the Step 2: Column Steps Definition page to create the data component.

Updating data components

Use the following procedure to update a data component:

1. On the Data Component Definition page, click the **Update Component** icon for the data component that you want to update.
2. On the Step 1: Table Column Selection page, make any desired changes to the name, description, and table column selections, then click the **Continue** button.
3. On the Step 2: Column Steps Definition page, make any desired changes, then click the **Finish** button.

Removing data components

Use the following procedure to remove a data component:

1. On the Data Component Definition page, click the **Remove Component** icon for the data component that you want to remove.
2. Click **Yes** in response to the prompt to remove the data component.

Using the Data Inspector to View and Edit Tables

This chapter covers the following topics:

- About the Data Inspector and Data Inspector Rules
- Creating Data Inspector Rules
- Viewing Data Inspector Rules
- Running Data Inspector Rules
- Duplicating Data Inspector Rules
- Updating Data Inspector Rule Names and Descriptions
- Updating Selection Settings for Data Inspector Rules
- Deleting Data Inspector Rules

About the Data Inspector and Data Inspector Rules

The Data Inspector is a facility that allows you to view tables and make minor adjustments to the data that they contain. The Data Inspector is not intended for large data entry tasks or for making extensive changes to data in tables.

You view and edit tables through the Data Inspector by creating rules. For each Data Inspector rule, there is an associated rule version that specifies a set of column selection characteristics for a given table. There is only one version for each Data Inspector rule.

By running a Data Inspector rule version, you can view the data in the columns specified by the definition for the version. If you have specified that one or more columns are editable in the version definition, you can also edit the data in those columns.

You can include a condition in the definition for a version. For example, you could specify a condition that selects only those rows from the FEM_BALANCES table where the balance is greater than 1,000. When you run a rule that includes this condition, only the rows in that table where the balance is greater than 1,000 will be displayed. For further information about conditions, see *About Conditions*, page 4-1.

Note: It is possible to create a data inspector rule where some required columns are not included. This allows users to view data, but will not allow the creation of new data or, in certain cases, the updating of data. The required columns ensure the uniqueness of data.

The Data Inspector Rules page is the starting point from which you can work with Data Inspector rules and versions.

For more information, see the following topics:

- Creating Data Inspector Rules, page 5-2
- Viewing Data Inspector Rules, page 5-3
- Running Data Inspector Rules, page 5-3
- Duplicating Data Inspector Rules, page 5-4
- Updating Data Inspector Rule Names and Descriptions, page 5-4
- Updating Selection Settings for Data Inspector Rules, page 5-4
- Deleting Data Inspector Rules, page 5-5

Creating Data Inspector Rules

Use the following procedure to create a new Data Inspector rule and version:

1. On the Data Inspector Rules page, click the **Create** button.
2. On the Step 1: Select Table page, do the following:
 1. In the Folder box, select the folder in which you want to save the rule.
 2. In the Data Inspector Name box, specify the name for the rule. The name you specify is also used for the rule version.
 3. If you want to provide a text description for the rule, type the description in the Description box.
 4. Specify the access rights for the rule. If you want all users other than yourself to have read-only access rights, so that they can view and run the rule but cannot update or delete it, select **Read**. If you want all other users to have read and write access rights, so that they are able to view, run, update, and delete the rule, select **Read/Write**.
 5. In the Table box, select the table that you want to access through the rule.
Only tables that have been registered with Enterprise Performance Foundation are available for selection. For information about registering tables, see Performing Table and Column Registration Tasks, page 10-1.
 6. If you want to include a condition in the rule definition, select the condition that you want to use in the Condition Rule box.
 7. Click the **Continue** button to display the Step 2: Select Columns page.
3. On the Step 2: Select Columns page, do the following:
 1. Select the columns that you want to display by moving them from the Available Columns box to the Columns Displayed (In Order) box.
Columns denoted by an asterisk (*) are required columns; if you attempt to create a new row when you run the rule, you will only be able to do so if you have selected all of the required columns in this step.
 2. Specify the order in which you want to display the selected columns when you run the rule. The column at the top of the list in the Columns Displayed (In

Order) box will appear as the farthest column to the left when you run the rule version; the column at the bottom will appear as the farthest column to the right.

To move a column in the Columns Displayed (In Order) box, thereby changing its placement in the display order, select the column and use the up and down arrow buttons in the Columns Displayed (In Order) box to move the column to the desired position.

Tip: Select the ledger to be the first column displayed, because the ledger selection controls the value sets of all remaining columns on the selected table. When the ledger is selected, all other dimension columns are reset, so the ledger should be selected first.

3. If you want to make one or more columns editable, so that you can change the values in them when you run the rule version, click the **Enable Editable** button and select the columns that you want to be editable on the Enable Editable Columns page. Note that not all columns can be designated as editable.
4. Use the Sort Settings section to order the data that the Data Inspector displays when you run a Data Inspector rule. You can specify as many as three columns on which to sort.

For example, if you have specified the Calendar Period column in the Balances table as the First Sort column name in ascending order, the Balances table will be sorted by calendar period in ascending order when you run the rule.

5. Click the **Finish** button to complete the rule and version definition.

Viewing Data Inspector Rules

Use the following procedure to search for and display existing Data Inspector rules on the Data Inspector Rules page:

1. In the Folder box, select the folder in which you want to search for rules.
2. If you want to refine the search criteria by name, specify all or part of a rule name.
3. Click the **Go** button to display a list of the rules and versions that meet the search criteria.

You can click on the name of a Data Inspector rule or version in the list to display information about the rule or version.

Running Data Inspector Rules

To run a Data Inspector rule, you run the rule version associated with the rule. Use the following procedure to run a version of a Data Inspector rule:

1. On the Data Inspector Rules page, click the **Run** icon for the rule version that you want to run.

When the rule version runs, the display shows the following:

- All of the rows in the table specified by the rule
- All columns that have been selected through the by the rule version definition
- The contents of each cell in the displayed rows and columns

2. If you want to change a value in an editable column (as specified by the rule version definition), enter the new value directly or use the **Search** icon to select a new value.
3. If you want to add a new row to the table, click the **Add Another Row** button. You can then enter values in any editable columns in the new row. If no columns have been specified as editable or not all of the required columns for the table were selected in the rule version definition, the **Add Another Row** button does not appear.
4. To delete rows from the table, select any rows that you want to delete and click the **Delete** button.
5. When you have finished making any changes to the table, click the **Apply** button to save the changes, or click the **Cancel** button if you do not want to save the changes. If you have not made any changes to the table, click either **Apply** or **Cancel** to close the display.

Duplicating Data Inspector Rules

You can duplicate a Data Inspector rule version to create a new rule and version as follows:

1. On the Data Inspector Rules page, display the version that you want to duplicate.
2. Click the **Duplicate** icon for the version.
3. On the Duplicate Business Object page, do the following:
 1. Select the folder in which you want to create the new rule and version.
 2. Specify the name for the new rule. The name you specify is also used for the rule version.
 3. If you want to provide a text description for the rule, type the description in the Description box.
 4. Click the **Apply** button.

The new rule and version appear on the Data Inspector Rules page.

Updating Data Inspector Rule Names and Descriptions

Use the following procedure to update the name and description for a Data Inspector rule:

1. On the Data Inspector Rules page, display the rule that you want to update.
2. Click the **Update** icon for the rule.

An update page for the rule appears.

3. Make any desired changes to the name and description.
4. Click the **Apply** button.

Updating Selection Settings for Data Inspector Rules

To update the selection settings for a Data Inspector rule, you update the rule version, as follows:

1. On the Data Inspector Rules page, display the rule version that you want to update.

2. Click the **Update** icon for the version.
3. On the Update Data Inspector page, do any of the following, as desired:
 - Make any desired changes to the columns that you want to display by adding columns to or removing columns from the Columns Displayed (In Order) box.

Columns denoted by an asterisk (*) are required columns; if you attempt to create a new row when you run the rule, you will only be able to do so if you have selected all of the required columns in this step.
 - If you want to change the order in which to display the columns when you run the rule, change the order in which the columns are listed in the Columns Displayed (In Order) box. The column at the top of the list in the Columns Displayed (In Order) box will appear as the farthest column to the left when you run the rule version; the column at the bottom will appear as the farthest column to the right.

To move a column in the Columns Displayed (In Order) box, thereby changing its placement in the display order, select the column and use the arrow buttons to move it to the desired position.
 - If you want to update the settings for which columns are editable and which are not, click the **Enable Editable** button and specify the desired settings Enable Editable Columns page.
 - If you want to update the column sort order, use the Sort Settings section to specify the order for the data that the Data Inspector displays when a you run a Data Inspector rule. You can specify as many as three columns on which to sort.
4. Click the **Apply** button.

Deleting Data Inspector Rules

To delete a Data Inspector rule, you delete the version for that rule. Use the following procedure to delete a rule version:

1. On the Data Inspector Rules page, click the **Delete** icon for the version that you want to delete.
2. Click **Yes** in response to the prompt to delete the version.

Working with Rule Sets

This chapter covers the following topics:

- About Rule Sets
- Creating Rule Sets
- Viewing Rule Sets
- Running Rule Sets
- Duplicating Rule Sets
- Updating Rule Set Names, Descriptions, and Access Settings
- Updating Rule Set Definition Settings
- Deleting Rule Sets

About Rule Sets

Rule sets are groupings of rules that allow you to run multiple rules together as a single operation. You can specify which rules you want to include in a rule set, and you can specify the order in which you want to run those rules when you run the rule set. You can also include other rule sets within a rule set.

Through the Rule Sets page, you can create, view, run, duplicate, export, and migrate rule sets.

For more information, see the following topics:

- Creating Rule Sets, page 6-1
- Viewing Rule Sets, page 6-2
- Running Rule Sets, page 6-2
- Duplicating Rule Sets, page 6-3
- Updating Rule Set Name, Descriptions, and Access Settings, page 6-3
- Updating Rule Set Definition Settings, page 6-3
- Deleting Rule Sets, page 6-4

Creating Rule Sets

Use the following procedure to create a rule set:

1. On the Rule Sets page, click the **Create** button.
2. On the Create Rule Set page, do the following:
 1. In the Folder box, select the folder in which you want to create the rule set.
 2. In the Rule Set Name box, specify the name for the rule set.
 3. In the Business Rule Type box, select the rule type for the rule set.
 4. If you want to provide a text description for the rule set, specify the description in the Description box.
 5. Specify the access rights for the rule set. If you want all users other than yourself to have read-only access rights to the rule set, select **Read**. If you want all other users to have read and write access rights to the rule set, select **Read/Write**.
 6. Click the **Continue** button.
3. On the Define Rule Set page, do the following:
 1. In the Available Rules and Rule Sets box, list the available rules and rule sets for the desired folder and category by selecting a folder in the Folder box and selecting a category in the Category box.
 2. If you want to filter the list of available rule sets by name, specify all or part of a name in the Name box in the Available Rules and Rule Sets box.
 3. Click the **Go** button in the Available Rules and Rule Sets box to display a list of rules and rule sets.
 4. Use the arrow buttons to move any rules and rule sets that you want to include in the new rule set from the Available Rules and Rule Sets box to the Selected Rules and Rule Sets box.
 5. Click the **Finish** button to create the rule set.

Viewing Rule Sets

Use the following procedure to search for and display existing rule sets on the Rule Sets page:

1. In the Folder box, select the folder in which you want to search for rule sets.
2. If you want to refine the search criteria by name, specify all or part of a rule set name.
3. Click the **Go** button to display a list of the rule sets that meet the search criteria.

You can click on the name of a rule set or version in the list to display information about the rule set or version.

Running Rule Sets

To run a rule set, you run the version associated with the rule set. Use the following procedure to run a version of a rule set:

1. On the Rule Sets page, click the **Run** icon for the rule set version that you want to run.
2. On the Execute Rule Set page, do the following:

- In the Effective Date box, specify the effective date for which you want to run the rule set. Enterprise Performance Foundation will run the appropriate version of the rule set, depending on the effective date that you specify.
 - In the Ledger box, specify the ledger in which you want the rule set to run.
 - In the Calendar Period box, specify the calendar period for the rule set. The calendar period, in conjunction with the data set group, specifies the input and output periods and data sets. Results are posted to the specified calendar period.
 - In the Dataset Group box, specify the data set group for the rule set. The data set group, in conjunction with the calendar period, specifies the input and output periods and data sets. Inputs are read from the specified data set group.
 - Select the **Continue Processing on Error** option if you want processing to continue if an error occurs. If you do not select this option, processing stops if an error occurs.
3. Click the **Submit** button to submit the rule set for processing.

Duplicating Rule Sets

You can duplicate a rule set version to create a new rule set and version as follows:

1. On the Rule Sets page, display the version that you want to duplicate.
2. Click the **Duplicate** icon for the version.
3. On the Duplicate Business Object page, do the following:
 1. Select the folder in which you want to create the new rule set and version.
 2. Specify the name for the new rule set. The name you specify is also used for the rule set version.
 3. If you want to provide a text description for the rule set, type the description in the Description box.
 4. Click the **Apply** button.

The new rule set and version appear on the Rule Sets page.

Updating Rule Set Names, Descriptions, and Access Settings

Use the following procedure to update the name, description, and access settings for a rule set:

1. On the Rule Sets page, display the rule set that you want to update.
2. Click the **Update** icon for the rule set.

An update page for the rule set appears.

3. Make any desired changes to the name, description, and access settings.
4. Click the **Apply** button.

Updating Rule Set Definition Settings

To update the definition settings for a rule set, you update the rule set version, as follows:

1. On the Rule Sets page, display the version that you want to update.
2. Click the **Update** icon for the version to display the update page, which lists all of the rules that are members of that version.
3. On the update page, do any of the following, as desired:
 - Enable or disable rules by selecting or deselecting the Enabled box associated with each rule. Enabled rules are included in processing when the rule set runs; disable rules are not included in processing.
 - Make any changes to the sequence in which the rules are processed when the rule set runs by changing the settings in the Sequence boxes.
 - To add or remove rules, click the Update Members button to display the Define Rule Set page. For further information about the Define Rule Set page, see *Creating Rule Sets*, page 6-1.
 - To run a validity check on the rule version definition, click the Validate button to display the Validate Rule Set page, through which you can submit a request to validate the rule set and post the results to a report.
4. Click the **Apply** button.

Deleting Rule Sets

To delete a rule set, you delete the version for that rule set. Use the following procedure to delete a rule set version:

1. On the Rule Sets page, click the Delete icon for the version that you want to delete.
2. Click **Yes** in response to the prompt to delete the version.

Working with Data Set Groups

This chapter covers the following topics:

- About Data Set Groups
- Creating Data Set Groups
- Viewing Data Set Groups
- Duplicating Data Set Groups
- Updating Data Set Groups
- Deleting Data Set Groups

About Data Set Groups

The Dataset dimension provides a means for categorizing data into groups, or *data sets*, that represent different types of data. For example, you could create data sets named Budget, Actual, and Forecast, then load all of the data of each of those types into the corresponding data set. Data sets thus provide a means to organize data.

Data set groups allow you to specify the input and output data sets used in calculations.

A data set group consists of one or more input data sets and one output data set. When you run a rule, you select a data set group; the data set group specifies the input data for the rule, and also specifies where the output data is to be stored after the rule has run.

For each data set group, there is an associated version, which is where you specify the input and output data sets for the data set group. There is only one version for each data set group.

You must specify one of the following for each input data set in a data set group:

- A specific calendar period that is related to the ledger that you have selected in the application preferences (as specified through the **Application Preferences** link in the global toolbar area).

The calendar period must be one that is part of a specific calendar, as defined by the ledger. As an example of a specific calendar period, you could specify the period jan-05 for an input data set.

- A calendar period that is an offset from the input date that is supplied when the calculation runs.

The offset can be a certain number of calendar periods in the future, or in the past, as compared to the input date for the calculation. You can also specify the type of calendar period for the offset, such as days, weeks, or months.

Some usage examples of offsets are as follows:

- You could specify the period that is one month after the input date specified for the calculation as an offset to ensure that you always use the previous month's results for the current month's processing.
- You might have statistics saved in a previous month's rows in a particular data set, and you could reference both the data rows and the statistic rows in one data set group.
- In a case where statistics might be recorded only once per year and stored in a particular calendar period in a particular data set, you could refer to that input data set in a row in the data set group and reference the exact calendar period that contains the statistic values.

When a data set group is used in a calculation and the data set group uses a relative offset to specify an input calendar period, the base date used to calculate the relative offset is the calendar period specified as a run parameter. For example, if the data set group includes a relative offset of -1 for Month and the Calendar Period is June 4, 2007, then the previous month (May) would be used as the input calendar period for the calculation.

A special case is an offset of zero (0), which indicates that there is no offset and that the current calendar period specified in the run parameters is to be used. This is useful in creating reusable or "generic" data set groups, as there is no need to update the calendar period each time a calculation is run with a particular data set group.

For more information, see the following topics:

- Creating Data Set Groups, page 7-2
- Viewing Data Set Groups, page 7-3
- Duplicating Data Set Groups, page 7-3
- Updating Data Set Groups, page 7-4
- Deleting Data Set Groups, page 7-5

Creating Data Set Groups

Use the following procedure to create a new data set group and version:

1. On the Dataset Groups page, click the **Create Dataset Group** icon.
2. On the Create Dataset Group: Step 1: Dataset Group Details page, do the following:
 1. In the Folder box, select the folder in which you want to save the data set group.
 2. In the Dataset Group Name box, specify the name for the data set group. The name you specify is also used for the associated version.
 3. If you want to provide a text description for the data set group, type the description in the Description box.
 4. Specify the access rights for the data set group. If you want all users other than yourself to have read-only access rights, so that they can view the group but cannot duplicate, update, or delete it, select **Read**. If you want all other users to have read and write access rights, so that they are able to view, duplicate, update, and delete the group, select **Read/Write**.

5. Click the **Continue** button to display the Create Dataset Group: Step 2: Dataset Group Definition page.
3. On the Create Dataset Group: Step 2: Dataset Group Definition page, do the following:
 1. In the Output Dataset box, select the desired output data set.

The data set that you specify as the output data set is automatically defined as an input data set, which ensures that all data currently in the specified output data set is preserved.
 2. Specify any additional input data sets. To specify an additional data set, click the **Add Another Row** button.

For each input data set, you must either select a specific calendar period in the Calendar Period box or select the Offset box, select the desired period type for the offset in the Period Type box, and specify the desired offset in the Period Offset box.

If you are using an offset, specify a number in the Period Offset box that defines the desired number of offset periods for the type selected in the Period Type box. Use a positive number to specify an offset subsequent to the input date for a calculation, and use a negative value to specify an offset prior to the calculation date. You can also specify an offset of zero (0), which indicates that the calendar period specified in the run parameters is to be used.

For example, to specify an offset that is two months after the calendar period for a calculation, select **Month** in the Period Type box and specify 2 in the Period Offset box; to specify an offset that is two months prior to the input date for a calculation, select **Month** in the Period Type box and specify -2 in the Period Offset box.

If you want to use the same period type as the type specified for Output Calendar Period when a calculation is run, select **Same as Output Calendar Period**.
 3. After you have defined all of the desired input data sets, click the **Finish** button to complete the data set group and version definition.

Viewing Data Set Groups

Use the following procedure to search for and display existing data set groups on the Data Set Groups page:

1. In the Folder box, select the folder in which you want to search for data set groups.
2. If you want to refine the search criteria by name, specify all or part of a data set group name.
3. Click the **Go** button to display a list of the data set groups and versions that meet the search criteria.

You can click on the name of a data set group or version in the list to display information about the group or version.

Duplicating Data Set Groups

You can duplicate a data set group version to create a new group and version as follows:

1. On the Data Set Groups page, display the version that you want to duplicate.
2. Click the **Duplicate** icon for the version.
3. On the Duplicate Business Object page, do the following:
 1. Select the folder in which you want to create the new group and version.
 2. Specify the name for the new group. The name you specify is also used for the associated version.
 3. If you want to provide a text description for the group, type the description in the Description box.
 4. Click the **Finish** button.

The new data set group and version appear on the Dataset Groups page.

Updating Data Set Groups

You can update the name, description, and input data sets for a data set group. Note that you cannot change the output data set for an existing data set group.

Use the following procedure to update a data set group:

1. On the Data Set Groups page, display the data set group version that you want to update.
2. Click the **Update** icon for the version.
3. On the on the Update Dataset Group page, do any of the following, as desired:
 - To update the data set group name or description, make the appropriate changes in the Dataset Group Name or Description boxes.
 - To add an additional input data set to the data set group, click the **Add Another Row** button.

For each input data set, you must either select a specific calendar period in the Calendar Period box or select the Offset box, select the desired period type for the offset in the Period Type box, and specify the desired offset in the Period Offset box.

If you are using an offset, specify a number in the Period Offset box that defines the desired number of offset periods for the type selected in the Period Type box. Use a positive number to specify an offset subsequent to the input date for a calculation, and use a negative value to specify an offset prior to the calculation date.

For example, to specify an offset that is two months after the input date for a calculation, select **Month** in the Period Type box and specify 2 in the Period Offset box; to specify an offset that is two months prior to the input date for a calculation, select **Month** in the Period Type box and specify -2 in the Period Offset box.

If you want to use the same period type as the type specified for Output Calendar Period when a calculation is run, select **Same as Output Calendar Period**.

- To delete an input data set from the data set group, select the input data set that you want to delete, click the **Delete** button, and click **Yes** in response to the prompt.

4. Click the **Apply** button to save the updates you have made.

Deleting Data Set Groups

To delete a data set group, you delete the associated version for that group. Use the following procedure to delete a data set group version:

1. On the Data Set Groups page, click the **Delete** icon for the version that you want to delete.
2. Click **Yes** in response to the prompt to delete the version.

Working with Requests

This chapter covers the following topics:

- About Concurrent Programs and Requests
- Submitting Requests
- Programs
- Viewing Requests

About Concurrent Programs and Requests

Concurrent programs are programs that can run without requiring continuous interaction on the part of the user.

To run a concurrent program, you submit a request using the Oracle Applications Standard Request Submission feature, which is available through Enterprise Performance Foundation. Enterprise Performance Foundation also allows you to view existing requests.

For further information about working with requests, see the following topics:

- Submitting Requests, page 8-1
- Programs, page 8-3
- Viewing Requests, page 8-9

For further information about submitting and monitoring requests in Oracle Applications, see *Oracle Applications User's Guide*.

Submitting Requests

This topic describes the process for using the Schedule Request pages to submit requests to run concurrent programs.

To begin, navigate to the Schedule Request: Name page.

The steps in the request submission process are as follows:

- Step 1: Name, page 8-2
- Step 2: Parameters, page 8-2
- Step 3: Schedule, page 8-2
- Step 4: Notifications, page 8-3

- Step 5: Printing, page 8-3
- Step 6: Review, page 8-3

Step 1: Name

On the Schedule Request: Name page, do the following:

1. In the Program Name box, select the program for which you want to submit a request.
2. If you want to provide a name for the request, specify the name in the Request Name box
3. Click the **Next** button.

Step 2: Parameters

On the Schedule Request: Parameters page, specify parameter values for the request, then click the **Next** button. See Programs, page 8-3 for parameter descriptions.

Not all programs require a set of parameters. If the program for which you are submitting a request does not require parameters, proceed to the next step in this procedure.

Step 3: Schedule

On the Schedule Request: Schedule page, you can use the simple schedule form as follows:

1. In the Schedule section, select either **As soon as possible** or **Start at specific date and time**, depending on when you want to submit the request. If you select **Start at specific date and time**, specify the start date and start time for submitting the request.
2. In the Recurrence section, select **Never repeat** if you want to submit the request only once, or select **Repeat** if you want to submit the request more than once.

If you select **Repeat**, specify the interval at which you want the request submission to repeat, along with the date and time as of which you want the submissions to end. If you do not specify an end date, the request continues to be resubmitted until you cancel it.

3. In the Schedule Name section, select the name of a schedule.

Alternatively, you can click the **Advanced Schedule** button to display the advanced version of the schedule form for the request. You can then specify the start date and start time for the request, the end date and end time for the request, and the months and days on which you want to submit the request

Select **Increment Date Parameters** if your request contains date parameters and you want to have the value for the parameter adjusted to match the resubmission interval. For example, if the value for the parameter is 01-Apr-2005 07:00 and your interval is monthly, the parameter is adjusted to 01-May-2005 07:00 for the next submission.

When you have finished your work on the Schedule Request: Schedule page, click the **Next** button.

Step 4: Notifications

On the notifications, page, specify the people whom you want to notify regarding the request. For each recipient, you can select the types of notifications that are generated.

To add additional recipients, use the **Add Another Row** button to add additional rows. To delete a recipient, click the Select box for that recipient and click the **Delete** button.

When you have finished your work on the Schedule Request: Notifications page, click the **Next** button.

Step 5: Printing

On the Schedule Request: Printing page, do the following:

1. In the Print Style box, select the style that you want to use. The choices that are available depend upon the program for which you are submitting the request.
2. In the Printer box, select the printer that you want to use to print copies of the request.
3. In the Copies box, specify the number of copies that you want to print. If you do not want to print any copies, specify 0 in the Copies box.
4. If you want to save the request information to a file, select **Save Output Files**.
5. Click the **Next** button.

Step 6: Review

On the Schedule Request: Review page, review the specifications for the request. If you want to change anything, use the **Back** button to return to the appropriate page; otherwise, click the **Submit** button to submit the request.

Programs

This topic describes the following programs, which are available through the Schedule Request pages:

- Detail Client Data Loader, page 8-4
- Dimension Hierarchy Loader, page 8-4
- Dimension Hierarchy Migration, page 8-5
- Dimension Member Loader, page 8-5
- Dimension Member Migration, page 8-6
- Executed Rule Removal, page 8-6
- Executed Rule Request Removal, page 8-7
- External General Ledger Data Loader, page 8-7
- Sign Extended Account Types, page 8-7
- Program - Enterprise Performance Foundation Discoverer Hierarchy Transformation Engine, page 8-8
- Program - Enterprise Performance Foundation General Ledger Balances Rule Engine, page 8-8

- Program - Enterprise Performance Foundation General Ledger Calendar Rule Engine, page 8-8
- Program - Enterprise Performance Foundation General Ledger Dimension Rule Engine, page 8-8
- Program - Enterprise Performance Foundation General Ledger Hierarchy Rule Engine, page 8-9

Detail Client Data Loader program

The purpose of the Detail Client Data Loader program is to load client data into tables (such as FEM_CHECKING, FEM_COLLATERAL, and the FEM_DATA1 through FEM_DATA20 tables) in Enterprise Performance Foundation.

The following table describes the parameters for the Detail Client Data Loader program.

Parameter	Description	Required?
Rule	Specifies the table to which the data is to be loaded.	Yes
Execution Mode	Describes the type of data load. Error Reprocessing is for a previous load that ended in error. Snapshot is for a "point in time" data load.	Yes
Ledger	Specifies the ledger under which the data is to be loaded.	Yes
Calendar Period	The calendar period for which you are loading data, filtered by the selected ledger.	Yes
Dataset	The name of the data set into which you are loading data.	Yes
Source System	Defines an identifier that can be used to group loaded rows together.	Yes

For further information about the Detail Client Data Loader program, see Detail Client Data Loader, page A-88.

Dimension Hierarchy Loader program

The purpose of the Dimension Hierarchy Loader program is to load hierarchies from interface tables that contain the hierarchy definitions.

The following table describes the parameters for the Dimension Hierarchy Loader program.

Parameter	Description	Required?
Rule Name	Specifies the load as a Hierarchy Load.	Yes
Execution Mode	Describes the type of data load. Error Reprocessing is for a previous load that ended in error. Snapshot is for a "point in time" data load.	Yes
Dimension	The dimension for which the hierarchy is to be loaded.	Yes
Hierarchy Name	The hierarchy object name that is to be loaded.	Yes
Hierarchy Definition Name	The hierarchy definition that is to be loaded.	Yes

For further information about the Dimension Hierarchy Loader program, see Dimension Hierarchy Loader, page A-52.

Dimension Hierarchy Migration program

The Dimension Hierarchy Migration program migrates a dimension hierarchy from the source system to the target system. You must run the program from the target system.

The following table describes the parameters for the Dimension Hierarchy Migration program.

Parameter	Description	Required?
Database Link	The database link that connects the source system to the target system.	Yes
Dimension	The dimensions whose members are to be migrated. The program moves the dimension members one dimension at a time.	Yes
Hierarchy	The name of the hierarchy to be migrated.	Yes
Hierarchy Version	The name of the hierarchy version to be migrated. If no version is specified, all versions will be migrated.	No
User Dimension Name	User-configurable dimension name from the source system. (For example, to populate User Configurable Dimension 1 in the target system with data from User Configurable Dimension 8 in the source system, specify User Configurable Dimension 8 for User Dimension Name.)	No

Dimension Member Loader program

The purpose of the Dimension Member Loader program is to load dimensions from interface tables that contain the dimension definitions.

The following table describes the parameters for the Dimension Member Loader program.

Parameter	Description	Required?
Execution Mode	Describes the type of data load. Error Reprocessing is for a previous load that ended in error. Snapshot is for a "point in time" data load.	Yes
Dimension	The dimension that is to be loaded	Yes

For further information about the Dimension Member Loader program, see Dimension Member Loader, page A-7.

Dimension Member Migration program

The Dimension Member Migration program migrates all the dimension member and attribute information for a dimension from the source system to the target system. You must run the program from the target system.

The following table describes the parameters for the Dimension Member Migration program.

Parameter	Description	Required?
Source Database	The database link that connects the source system to the target system.	Yes
Dimension	The dimensions whose members are to be migrated. The program moves the dimension members one dimension at a time.	Yes
User Dimension Name	User-configurable dimension name from the source system. (For example, to populate User Configurable Dimension 1 in the target system with data from User Configurable Dimension 8 in the source system, specify User Configurable Dimension 8 for User Dimension Name.)	No

Executed Rule Removal program

The purpose of the Executed Rule Removal program is to remove executed rules in cases where you know the object ID and the request ID of the executed rule.

The following table describes the parameters for the Executed Rule Removal program.

Parameter	Description	Required?
Object	The business rule description for the request ID	No
Request ID	The request ID for the executed rule	Yes
Include Dependencies	If the request has chained dependencies, this option will undo them as well.	Yes
Ignore Dependency Errors	If the request generates errors in dependencies, this option will determine whether the program continues processing or stops when an error is encountered.	Yes

Executed Rule Request Removal program

The purpose of the Executed Rule Request Removal program is to remove executed rules in cases where you know only the request ID of the executed rule.

The following table describes the parameters for the Executed Rule Request Removal program.

Parameter	Description	Required?
Request ID	The request ID for the executed rule request	Yes
Include Dependencies	If the request has chained dependencies, this option will undo them as well.	Yes
Ignore Dependency Errors	If the request generates errors in dependencies, this option will determine whether the program continues processing or stops when an error is encountered.	Yes

External General Ledger Data Loader program

The purpose of the External General Ledger Data Loader program is to load data from interface tables that contain definitions for data from a general ledger other than Oracle General Ledger.

The following table describes the parameters for the External General Ledger Data Loader program.

Parameter	Description	Required?
Execution Mode	Describes the type of data load. Error Reprocessing is for a previous load that ended in error. Snapshot is for a "point in time" data load.	Yes
Ledger	Specifies the ledger under which the data is to be loaded	Yes
Calendar Period	The calendar period for which you are loading data, filtered by the selected ledger	Yes
Budget	List of budgets to assign to the data load	No
Encumbrance Type	List of encumbrance types to assign to the data load	No
Dataset	The name of the data set into which you are loading data	Yes
Period Specific Amounts	Specifies which of the PTD, QTD, and YTD data columns that you want to populate	Yes

For further information about the Import External GL Data program, see External General Ledger Data Loader, page A-72.

Sign Extended Account Types program

The purpose of the Sign Extended Account Types program is to assign signage to extended account types. Before performing any data loading operations or calculations, do the following:

1. Set the FEM: Signage Methodology profile option as appropriate for your installation.

2. Run the Sign Extended Account Types program.

There are no parameters for the Sign Extended Account Types program.

Program - Enterprise Performance Foundation Discoverer Hierarchy Transformation Engine program

The Discoverer Hierarchy Transformation Engine program is available with the Enterprise Performance Foundation Administrator user responsibility. The purpose of this program is to transform a hierarchy from Enterprise Performance Foundation into the format required by Oracle Discoverer. The program transforms all hierarchies for a specified dimension or for all dimensions and places the results in the "EPF - Core Business Area" business area. You must install the Discoverer End User Layer before running this program.

The following table describes the parameters for the Discoverer Hierarchy Transformation Engine program:

Parameter	Description	Required?
Dimension Name	Specifies the dimension for the hierarchies to be transformed. Specify a specific dimension for the hierarchies to be transformed, or specify All for all dimensions.	Yes

Program - Enterprise Performance Foundation General Ledger Balances Rule Engine program

The purpose of the General Ledger Balances Rule Engine program is to allow users (administrators) to migrate data from Oracle General Ledger to Enterprise Performance Foundation through the use of business rules. The rules offer a way to specify and store various settings, such as the types of balances to load from each ledger and the amount type and translated balances information. Ultimately, they also offer a mechanism for scheduling balance transfer.

For information about using the General Ledger Balances Rule Engine program, see *Execute Balances Rules*, page 12-27.

Program - Enterprise Performance Foundation General Ledger Calendar Rule Engine program

The purpose of the General Ledger Calendar Rule Engine program is to transfer an Oracle General Ledger calendar into Enterprise Performance Foundation.

For information about using the General Ledger Calendar Rule Engine program, see *Calendars*, page 12-16.

Program - Enterprise Performance Foundation General Ledger Dimension Rule Engine program

The purpose of the General Ledger Dimension Rule Engine program is to populate Enterprise Performance Foundation with detail dimension members based on the selected dimension rule. Such dimensional information is then used in transferring Oracle General Ledger balances for reporting and analytical processing.

For information about using the General Ledger Dimension Rule Engine program, see *Execute Dimension Rules*, page 12-25.

Program - Enterprise Performance Foundation General Ledger Hierarchy Rule Engine program

The purpose of the General Ledger Hierarchy Rule Engine program is to populate Enterprise Performance Foundation with hierarchical information specified in the selected hierarchy rule. Although these hierarchies are not used during the Oracle General Ledger balances transfer process, they are vital to analytical processes such as consolidation and budgeting.

For information about using the General Ledger Hierarchy Rule Engine program, see *Execute Hierarchy Rules*, page 12-26.

Viewing Requests

Through the Requests page, you can display requests that you have submitted and view information about them. You cannot view requests submitted by other users.

Use the View box on the Requests page to select the requests that you want to view, as follows:

- All — Displays all requests
- Completed — Displays all completed requests
- Last 24 hours — Displays all requests submitted within the last 24 hours
- Pending — Displays all pending requests
- Running — Displays all requests that are currently running
- Search — Allows you to search for requests. Select **Search** in the View box and click the **Go** button to display a request search facility.

Removing Rule Execution Results

This chapter covers the following topics:

- About Rule Execution Results and Process Locking
- Viewing Rule Execution Results
- Removing Rule Execution Results and Process Locks

About Rule Execution Results and Process Locking

When you run a rule that performs a calculation, the system places a process lock on the rule version that generated the results. You cannot run that same rule version again until you delete the rule execution results and remove the process lock.

For example, suppose that you run a rule that performs a calculation that generates 10 rows. When you run the rule, the system places a process lock on the rule version; this prevents users from updating the rule, which could make it difficult or even impossible to understand the calculation details that produced the results.

However, if you decide that you have made a mistake, and you want to change the calculation and then run it again for the same data set and calendar period, you will be unable to do so because of the process lock. Before you can run the calculation again for the same data set and calendar period, you must first delete the rule execution results, which will remove both the updates to the database and the related process lock.

Through the Executed Rules page, you can view rule execution results, and you can also delete rule execution results and remove process locks. Note the following considerations regarding the Executed Rules page:

- The Executed Rules page displays execution results for all rules to which you have access. You must have a minimum of Read access to the folder that contains a particular rule to see execution results for that rule.
- The Executed Rules page displays execution results for all rules for which removal of execution results is currently in progress. You cannot select such rule executions for removal of execution results.
- The Executed Rules page displays execution results for all rules that have been submitted for removal of execution results but for which execution results were not removed. You can select such rule executions and make another attempt to remove execution results.
- The Executed Rules page does not display execution results for rules for which execution results have already been removed.

For more information, see the following topics:

- Viewing Rule Execution Results, page 9-2
- Removing Rule Execution Results and Process Locks, page 9-2

Viewing Rule Execution Results

On the Executed Rules page, use the Search By fields to specify the search criteria for the rule execution results that you want to view, then click the **Go** button to display the rule execution results that match your search criteria. You can also click the **Advanced Search** button to display the Advanced Search page, which allows you to specify additional search criteria.

The Executed Rules page provides general information about rule execution results. You can also view additional information, as follows:

- If you want to view detailed information about the formula for a rule, click the **Item Name** link for the rule to display the View Formula page.
- If you want to access the request for a rule, click the **Request ID** link for the rule to access the request in the concurrent manager user interface.

Removing Rule Execution Results and Process Locks

Use the following procedure to remove execution results and process locking for rules:

1. On the Executed Rules page, use the Search By fields to specify the search criteria for the rule execution results that you want to remove, then click the **Go** button to display the rule execution results that match your search criteria. You can also click the **Advanced Search** button to display the Advanced Search page, which allows you to specify additional search criteria.
2. Select the rule or rules for which you want to remove execution results and process locking and click **Undo**. If you want to remove execution results and process locking for all of the listed rules, click **Undo All**.

Note: If one or more of the selected rules or their dependents are not eligible for removal of execution results, you will not be able to remove execution results for any of the selected rules.

3. At the confirmation prompt, confirm that you want to remove the execution results. The system removes the execution results for the selected rules, along with the process locks for the associated rule versions.

Performing Administrative Tasks

This chapter covers the following topics:

- About Administrative Tasks in Enterprise Performance Foundation
- Performing Table and Column Registration Tasks
- Performing Dimension Administration Tasks
- Working with Value Sets
- Working with Global Value Set Combinations
- Working with Registered Database Links
- Working with Process Tuning Options
- Working with Security Folders
- Customizing the Home Page

About Administrative Tasks in Enterprise Performance Foundation

Through the administration facility, users can perform the administrative and setup functions that allow them to customize the system to best suit their needs.

Many of the pages in the administration facility provide a **Printable Page** button, which allows you to produce a version of the page that is formatted for printing. To produce and print pages using this feature, do the following:

1. Click the **Printable Page** button.
2. Use your browser's print function to print the formatted page.
3. Use your browser's back button to return to the previous page.

Performing Table and Column Registration Tasks

The core metadata for Enterprise Performance Foundation consists of table and column definitions and properties. Before tables and columns can be recognized and used by related applications, these tables and columns must be defined and registered within the system. Through the table and column registration facility, you can register tables and columns for use by user interfaces and concurrent programs.

Prior to using the table and column registration facility in Enterprise Performance Foundation, the database administrator for the E-Business Suite must do the following:

1. Determine which tables and columns are to be used for processing.

2. For each table in the system, there must be a unique index and a related interface table. It is necessary to alter the unique index for each table to include the columns to be used for processing. In addition, it is necessary to set each column included in the unique index to NOT NULL on the physical table; this must be done for both the base data table and the interface table.

After the unique index has been set for each base data table and interface table, the database administrator must use the Enterprise Performance Foundation table and column registration facility to do the following:

1. Register the new tables (this step applies only for tables that have not previously been registered).
2. Synchronize each of the base data tables.

Note: Synchronization is necessary whenever new tables are to be registered or there have been changes to a previously-registered table, and must be performed before defining or updating registration information.

3. Define the registration information for each of the base data tables (interface tables do not need to be registered).

Enterprise Performance Foundation supports the definition of user-defined tables. To be available for use, a user-defined table must have a synonym included in the APPS schema, and it must include the appropriate columns to be eligible for specific table classifications. Use the following SQL statement to define a synonym, where *USER_DEF_TABLE* represents the name of the table:

Example

```
CREATE SYNONYM APPS.USER_DEF_TABLE FOR FEM.USER_DEF_TABLE;
```

It is also possible to register fully updateable views.

With the exception of certain customer account tables, seeded tables cannot be modified. This means that you cannot add or remove columns from the FEM_BALANCES or the FEM_DATA1-20 tables.

For more information, see the following topics:

- Registering new tables, page 10-2
- Viewing table registrations, page 10-5
- Synchronizing table registrations, page 10-5
- Updating table display names and descriptions, page 10-6
- Updating table registrations, page 10-6
- Removing table registrations, page 10-6

Registering new tables

You can register tables through the Table Registration Wizard. Click the **Register New Object** button on the Table Registration page to open the Table Registration Wizard.

Important: You must complete all of the steps in the Table Registration Wizard to register a table. If you do not complete all of the steps, the table registration is incomplete, and the table may not be recognized by other applications.

The Table Registration Wizard consists of the following steps:

- Step 1: Register New Table, page 10-3
- Step 2: Table Column Display Names, page 10-3
- Step 3: Table Processing Key, page 10-4
- Step 4: Interface Table Column Mappings, page 10-4
- Step 5: Table Classifications, page 10-4
- Step 6: Table Column Property Assignments, page 10-5

Step 1: Register New Table

In the Register New Table step, do the following:

1. In the Schema box, use the search icon to select the schema that contains the table that you want to register.
2. In the Table Name box, use the search icon to select the name of the table that you want to register.
3. In the Table Display Name box, specify the display name for the table (the name for the table that is displayed application user interfaces).
4. In the Description box, specify a text description for the table.
5. Click the **Next** button to display the Table Column Display Names step

Step 2: Table Column Display Names

In the Table Column Display Names step, you can specify display names, data types, and descriptions for columns. Use the appropriate fields to specify these values, noting the following:

- The system might have locked the information in one or more fields for certain columns to protect the integrity of tables; such fields cannot be updated.
- If you want to enable a column for use in application user interfaces, select the **Enable** field for that column. If you do not select the **Enabled** box, the column will not be displayed in application user interfaces, and it will not be available for use in any business rules.
- If you want to change the data type, select the desired data type from the drop-down list under **CPM Data Type**. Depending on the column definition in the underlying database, different choices are available for **CPM Data Type**.
- If you select **Dimension** as the CPM data type for a column, then you must supply a value for **Dimension Name** by selecting the dimension associated with the column.
- If you select **Term**, **Frequency**, or **Statistic** as the CPM data type for a column, then you must supply a value for **UOM column display name** by making a selection from the list of values that associates another column with the item that describes the unit of measure for the term, frequency, or statistic.

When you have finished, click the **Save and Next** button to display the Table Processing key step.

Step 3: Table Processing Key

In the Table Processing Keys step, use the drop-down box to select the unique index that represents the processing key for the table.

When you have finished, click the **Save and Next** button to display the Interface Table Column Mappings step.

Step 4: Interface Table Column Mappings

For user-defined tables, you must specify the mappings between the columns in the base tables and the columns in the interface tables (the mappings are already defined for seeded tables). In the Interface Table Column Mappings step, specify the mappings for user-defined tables as follows:

1. In the Schema box, specify the schema that contains the interface table for which you want to provide column mappings.
2. In the Table Name box, specify the name of the interface table for which you want to provide column mappings.
3. Click the **Go** button to display a list of column names.
4. For each base table column listed in the Column Name field, specify the interface table column that you want to map to that base table column by selecting the appropriate value in the Interface Column Name field.
5. Click the **Save and Next** button to display the Table Classifications step.

Step 5: Table Classifications

Table classifications determine how tables are used in Enterprise Performance Foundation. Use the Table Classifications step to specify classifications for tables.

Move the classifications between the Available Classifications and Selected Classifications boxes as needed, so that only the classifications that you want to specify for the table are listed in the Selected Classifications box. You can move classifications between boxes by selecting one or more classifications and clicking the **Move** and **Remove** buttons, or you can move all classifications at once by using the **Move All** and **Remove All** buttons.

Note: The list in the Available Classifications box contains all of the valid table classifications. At a minimum, select the classifications that are pertinent to your purposes, based on the types of calculations to be performed on those tables. You must select at least one classification to successfully register the table, and additional classifications might be needed, depending on the features that will be used in the application.

If you want to specify a certain classification for a table but that classification does not appear in the Available Classifications box, do the following:

Note: For a table to qualify for the table classifications of ACCOUNT_PROFITABILITY @ FTP_CASH_FLOW, FTP_NON_CASH_FLOW and FTP_OPTION_COST, the processing key must only have the columns of CAL_PERIOD_ID, DATASET_CODE, SOURCE_SYSTEM_CODE, ID_NUMBER defined. If extra columns are included, the table will not qualify for these classifications.

1. Click the **Other Classifications** button to display the Other Classifications page, which provides a list of all classifications.
2. On the Other Classifications page, find the desired classification and expand the entry so that you can see which classification requirements are missing, thus preventing the table from appearing as an available classification.
3. Return to the Table Classifications step and complete the Table Registration Wizard.
4. Correct the problem by supplying any missing classification requirements.
5. Synchronize the table registrations.
For further information, see Synchronizing table registrations, page 10-5.
6. Run the Table Registration Wizard again and specify the desired classifications in the Table Classifications step.

When you have finished your work in the Table Classifications step, click the **Save and Next** button to display the Table Column Property Assignments page.

Step 6: Table Column Property Assignments

In the Property box in the Table Column Property Assignments step, do one of the following:

- Select **Mapping Input Property** if you want the columns in the table to be listed in the Initial Data Source section on the Update Formula page for mapping rules.
- Select **Mapping Output Property** if you want the columns in the table to be listed in the Debit and Credit sections on the Update Formula page for mapping rules.

Use the **Add Columns** button to add additional columns.

To delete a column, click the **Delete** icon for that column.

When you have finished, click the **Finish** button to complete the Table Registration Wizard.

Viewing table registrations

You can search for and display existing table registrations through the Table Registration page. To display an existing table registration, specify the name of the table in the Table Name box and click the **Go** button.

Synchronizing table registrations

Whenever you make any changes to a registered table, you must synchronize the registration for the table (that is, reread the table so that Enterprise Performance Foundation can recognize and incorporate the changes) before making any updates related to the table.

Use the following procedure to synchronize a table registration:

1. On the Table Registration page, display the table for which you want to synchronize the registration.
2. Click the **Synchronize** icon for the table.

Updating table display names and descriptions

Use the following procedure to update the display name and description for a table:

1. On the Table Registration page, display the table that you want to update.
2. Click on the name of the table to display the Update Table Display Name page.
3. Make any desired changes to the table name and description.
4. Click the **Apply** button to apply the updates.

Updating table registrations

Use the following procedure to update a table registration:

1. On the Table Registration page, display the table that you want to update.
2. Click the **Update** icon for the table to open the Update Table Registration Wizard.

The steps in the Update Table Registration Wizard are the same as Steps 2 through 6 in the Table Registration Wizard. For further information, see Registering new tables, page 10-2.

Removing table registrations

You can remove the registration for a table so that the table is no longer available in Enterprise Performance Foundation.

Important: If a rule refers to a table for which you remove the registration, the rule will no longer be valid.

Use the following procedure to remove a table registration:

1. On the Table Registration page, display the table for which you want to remove the registration.
2. Click the **Unregister** icon for the table, then choose **Yes** in response to the confirmation prompt.

Performing Dimension Administration Tasks

Through the Dimension Administration page, you can display a list of dimension maintenance tasks for a specified dimension. For each task, the Dimension Administration page shows when and by whom the last update was made, and indicates the status. The status icons are as follows:

- A solid square indicates that the task has not been completed or is in the system installation default state.
- A check mark indicates that the task has been completed.

For further information, see the following topic:

- Specifying dimension and column display names, page 10-6

Specifying dimension and column display names

You can make changes to the display names for dimensions and columns to make the names more user-friendly.

Use the following procedure to specify dimension and column display names:

1. In the Dimension box on the Dimension Administration page, select the dimension for which you want to specify display names and click the **Go** button.
2. Click the **Go To Task** icon for the Dimension Display Names task.
3. On the Dimension Column Display Names page, make any desired changes to the dimension display name and the column display names.
4. Click the **Apply** button to apply the changes that you have specified.

Specifying dimension properties

You can change the dimension properties for a dimension, as follows:

1. In the Dimension box on the Dimension Administration page, select the dimension for which you want to specify properties and click the **Go** button.
2. Click the **Go To Task** icon for the Dimension Properties task.
3. On the Dimension Properties page, make any desired changes to the dimension properties, as follows:
 - Select **Line** for the dimension type if you want to associate the dimension with particular predefined attributes in an application.

For example, if the dimension is to be used in Enterprise Planning and Budgeting and you want to be able to specify which members are to be loaded or defined in calculations in business processes, select **Line** as the dimension type (no more than one dimension can have **Line** as the dimension type within a single business area in Enterprise Planning and Budgeting).
 - Select **Other** for the dimension type if you do not need to associate the dimension with particular predefined attributes in an application.
4. Click the **Apply** button.

Working with Value Sets

There can be multiple lists of values for a dimension. By defining value sets, where each value set represents a particular list of values, you can distinguish these lists of value from one another. Value sets are useful when there are two systems that use the same dimensional identifier for different purposes.

For further information, see the following topics:

- Creating value sets, page 10-7
- Updating value sets, page 10-8
- Deleting value sets, page 10-8

Creating value sets

Use the following procedure to create a new value set:

1. On the Value Sets page, click the **Create** button.
2. On the Create Value Set page, specify the following:
 - A short identifier for the value set in the Code box

- The full name for the value set in the Value Set Name box
 - An optional text description for the value set in the Description box
 - The dimension to which the value set is to be assigned in the Dimension box. Click the search icon to search for and select a dimension.
3. Click the **Apply** button to create the value set.

Updating value sets

Use the following procedure to update an existing value set:

1. On the Value Sets page, search for the value set that you want to update. You can refine the search by value set name, dimension, or both. When you have specified the desired search values, click the **Go** button to display a list of value sets that meet the search criteria.
2. Click the **Update** icon for the value set that you want to update.
3. On the Update page, update the value set name and description, as desired.
4. Click the **Apply** button to apply the updates.

Deleting value sets

Use the following procedure to delete a value set:

1. On the Value Sets page, search for the value set that you want to delete. You can search by value set name, dimension, or both. When you have specified the desired search values, click the **Go** button to display a list of value sets that meet the search criteria.
2. Click the **Delete** icon for the value set that you want to delete. Note that some value sets cannot be deleted; the **Delete** icon is disabled for these value sets.
3. On the Delete page, confirm that you want to delete the value set.

Working with Global Value Set Combinations

After you have created value sets, you can create global value set combinations. A global value set combination is a defined combination of value sets in which there is one value set for each dimension in the system. Through a ledger attribute, global value set combinations are associated with ledgers for processing.

Through the Global Value Set Combinations page, you can manage global value set combinations.

For further information, see the following topics:

- Creating global value set combinations, page 10-8
- Updating global value set combinations, page 10-9
- Deleting global value set combinations, page 10-9

Creating global value set combinations

Use the following procedure to create a global value set combination:

1. On the Global Value Set Combinations page, click the **Create** button.

2. On the Create Global Value Set Combination page, specify the name for the new global value set combination and a value set for each dimension. For each dimension, you can use the default value set, or you can choose a different value set from the list in the Value Set box.
3. Click the **Apply** button to create the global value set combination.

Updating global value set combinations

Use the following procedure to update a global value set combination:

1. On the Global Value Set Combinations page, search for the global value set combination that you want to update. When you have specified the desired search value, click the **Go** button to display a list of global value set combinations that meet the search criteria.

For each global value set combination displayed in the Results section, you can view the component value sets by expanding the global value set combination. To expand a global value set combination, click the **Expand** icon immediately preceding the global value set combination.

2. Click the **Update** icon for the global value set combination that you want to update.
3. Make any desired changes to the global value set combination name. Note that you cannot make any changes to the value sets assigned to dimensions for existing global value set combinations.
4. Click the **Apply** button to update the global value set combination.

Deleting global value set combinations

You can delete a global value set combination, providing that the global value set combination has not been assigned to a ledger. Use the following procedure to delete a global value set combination:

1. On the Global Value Set Combinations page, search for the global value set combination that you want to delete. You can search by global value set combination name, dimension, or both. When you have specified the desired search values, click the **Go** button to display a list of global value set combinations that meet the search criteria.
2. Click the **Delete** icon for the global value set combination that you want to delete.
3. On the Delete page, confirm that you want to delete the global value set combination.

Working with Registered Database Links

About registered database links

A database link defines a communication path between a local database and a remote database (between a test instance and a production instance, for example). You must register a database link in Enterprise Performance Foundation before you can use the link to transfer rules and data between databases.

For further information, see the following topics:

- Viewing Registered Database Links, page 10-10

- Registering Database Links, page 10-10
- Updating Database Link Registrations, page 10-10
- Deleting Database Link Registrations, page 10-10

Viewing registered database links

To view a list of the database links that are currently registered, navigate to the Registered Database Links page.

Registering database links

Use the following procedure to register database links:

1. On the Registered Database Links page, click the **Register Additional Link** button.
2. On the Step 1: Select Link to Register page, do the following:
 1. Select the database link that you want to register.
 2. Click the **Continue** button.
3. On the Step 2: Assign Link Name page, do the following:
 1. In the Registered Name field, specify the name under which you want to register the database link.
 2. If you want to provide a text description for the link registration, specify the description in the Description box.
 3. Click the **Finish** button to complete the link registration.

Updating database link registrations

Use the following procedure to update a database link registration:

1. On the Registered Database Links page, click the **Update** icon for the link registration that you want to update.
2. On the Update Database Link page, do the following:
 1. Make any changes in the Registered Name and Description fields, as desired.
 2. Click the **Apply** button to apply the changes you have made.

Deleting database link registrations

Use the following procedure to delete a database link registration:

1. On the Registered Database Links page, click the **Delete** icon for the link registration that you want to delete.
2. Choose **Yes** in response to the confirmation prompt to delete the registration for the database link.

Working with Process Tuning Options

Through the Process Tuning Options page, you can work with sets of tuning options, or rules, that affect the manner in which the processing programs function. You can use

these rules to tune the performance of the programs, according to the options that you define for the rules or rule types.

For further information, see the following topics:

- Creating tuning options, page 10-11
- Updating tuning options, page 10-13
- Deleting tuning options, page 10-13

Creating tuning options

This topic describes the process for creating tuning options.

To begin, click the **Create** button on the Process Tuning Options page to display the Tuning Options: Assignment Level page, the first step in the process.

Step 1: Assignment level

On the Tuning Options: Assignment Level page, search for the do the following:

1. In the Rule Type box, the rule type for the tuning option.
2. In the Rule Step box, select one of the following:
 - **All** — Use this setting for rules that have only a single step.
 - **Incremental** — Sets parameters for incremental external General Ledger data integration rules (available only if the specified rule type is **External GL Data Integration Rule**).
 - **Snapshot** — Sets parameters for external General Ledger data integration rules (available only if the specified rule type is **External GL Data Integration Rule**).
 - **Generic Data Table** — Sets parameters for deleting the results of a data-loading operation in a generic data table (available only if the specified rule type is **Executed Rules**).
 - **FEM Balances** — Sets parameters for deleting the results of a data-loading operation in the FEM Balances table (available only if the specified rule type is **Executed Rules**).
3. If you want to assign the tuning option to a specific rule, specify the name of the rule in the Rule Name box. You can use the search icon to search for a rule name. If you want to apply the tuning option to all rules, leave the Rule Name box blank.
4. Click the **Continue** button to display the Update Tuning Options page.

Step 2: Update tuning options

On the Update Tuning Options page, click the **Update** button to display the Multiprocessing: Parameters page.

Step 3: Parameters

The Multiprocessing: Parameters page lists parameters and allows you to specify values for those parameters.

Note: Some of the parameter values described here may not be available for all rule types.

On the Multiprocessing: Parameters page, do the following:

1. In the Data Slice Type box, specify one of the following:
 - **Engine Specific** — Defined as a program default.
 - **Value Ranges** — Uses ranges of the specified column.
 - **Distinct Values** — Uses distinct values for the specified column.
2. In the Partition Code box, specify one of the following:
 - **No Partitions** — Indicates that the Enterprise Performance Foundation processing program fulfills unit of work requests regardless of any table partitioning. As each individual process completes, it requests the next unit of work segment, whether or not that segment belongs in the same table partition. Use this setting if you do not have Oracle Table Partitioning in your database.
 - **Shared Partitions** — Indicates that the Enterprise Performance Foundation processing program fulfills unit of work requests so that each process works against a specific partition unless it is idle. Idle processes then work against the next available unit of work segment, whether or not that segment belongs in the same table partition.
 - **Dedicated Partitions** — Indicates that the Enterprise Performance Foundation processing program fulfills unit of work requests so that each process works against a specific partition.
3. In the Number of processes box, specify the number of processes to run.

Note: For the Calendar Period dimension, the Number of Processes setting is limited to a value of 1, because setting a value greater than 1 could cause data integrity violations for Calendar Period dimension members. For any other dimension, you can increase the number appropriately for your system to take advantage of the multiprocessing capabilities of the loader.
4. If the Commit Frequency box is displayed and you want to specify how frequently data is saved to the database during processing, use the Commit Frequency box to specify the number of rows that are to be processed prior to each save operation.
5. If the Array Size Rows box is displayed and you want to specify the number of rows to be updated at a time, specify the number of rows in the Array Size Rows box.
6. Click the **Next** button to display the Multiprocessing: Data Slicing Columns page.

Step 4: Data Slicing Columns

Processing programs determine units of work for any job based on the process data slicing code, which is comprised of one or more columns by which data in the processed table is segmented. The individual segments are the defined units of work. The multiprocessing model enables you to specify different unit of work definitions for your processes.

Note: You can either divide the process by the number of slices for all the data in the process or process a given number of rows per slice. Click the box next to the desired Data Slicing Option to select which method the process uses.

1. On the Multiprocessing: Data Slicing Columns page, do one of the following:
 - Select the **Number of Slices** option and specify the number of slices in the Number of Slices box.
 - Select the **Rows per Slice** option and specify the number of rows per slice in the Rows per Slice box.
2. In Input Column field, specify the input column, depending on the rule type specified on the Tuning Options: Assignment Level page.
3. If you want to delete an input column, click the **Delete** icon for that column.
4. Click the **Apply** button to create the tuning option.

Updating tuning options

Use the following procedure to update tuning options:

1. On the Process Tuning Options page, search for the rule that you want to update. You can refine the search by rule type, rule step, rule name, or a combination thereof. When you have specified the desired search values, click the **Go** button to display a list of rules that meet the search criteria.
2. Click the **Details** icon for the rule that you want to update.
3. On the Update Tuning Options page, click the **Update** button.
4. On the Multiprocessing: Parameters page, make any desired updates to the settings for the rule, then click the **Next** button.

See Step 3: Parameters, page 10-11 for further information about the settings available on the Multiprocessing: Parameters page.

5. On the Multiprocessing: Data Slicing Columns page, make any desired updates to the settings for the rule. You can delete a column by clicking the **Delete** icon for that column.

See Step 4: Data Slicing Columns, page 10-12 for further information about the settings available on the Multiprocessing: Data Slicing Columns page.

6. Click the **Apply** button on the Multiprocessing Data Slicing Columns page to apply the updates that you have specified.

Deleting tuning options

Use the following procedure to delete tuning options:

1. On the Process Tuning Options page, search for the rule that you want to delete. You can refine the search by rule type, rule step, rule name, or a combination thereof. When you have specified the desired search values, click the **Go** button to display a list of rules that meet the search criteria.
2. Select the rule or rules that you want to delete by checking the Select box for each rule that you want to delete, then click the **Delete** button.

Working with Security Folders

Security folders provide security for objects in the system.

The system stores user-created objects in security folders. For each folder, the administrator of the folder (or any user that has system administrator access authorization) can grant users access rights of read, update, or no access to the folder and its contents.

Note: There are two seeded folders, Data Integration and Default. The Data Integration folder contains the rules for the loader programs, and a user must be assigned to this folder to be able to run the loaders. The Default folder contains the default calendar period hierarchy, and a user must be assigned to this folder to be able to update the default calendar period hierarchy for additional processing.

For more information, see the following topics:

- Creating Security Folders, page 10-14
- Viewing Security Folders, page 10-14
- Duplicating Security Folders, page 10-15
- Updating Security Folders, page 10-15
- Deleting Security Folders, page 10-15

Creating security folders

Use the following procedure to create a new security folder:

1. On the Security Folders page, click the **Create** button to display the Step 1: Create Folder page.
2. Specify the name for the new folder in the Folder Name box.
3. If you want to provide a text description for the folder, type the description in the Description box.
4. Do one of the following:
 - If you do not want to specify user assignments for the folder, click the **Finish** button to create the folder.
 - If you want to specify user assignments for the folder, click the **Continue** button to display the Step 2: Security Folder User Assignments page.

On the Step 2: Security Folder User Assignments page, specify user assignments as follows: For each user assignment, use the Responsibility and User boxes to specify the responsibility and user, then click the **Go** button to add the assignment to the list of users.

When you have completed all of the user assignments, click the **Finish** button to create the folder.

Viewing security folders

Use the following procedure to view existing security folders:

1. On the Security Folders page, use the Folder Name box to specify the desired search criteria to search for security folders by name.

2. Click the **Go** button to display a list of the security folders that meet the search criteria.

Duplicating security folders

Use the following procedure to duplicate a security folder:

1. On the Security Folders page, display the folder that you want to duplicate.
2. Click the **Duplicate** icon for the folder to display the Copy Folder page.
3. Specify the name for the duplicate copy of the folder in the Folder Name box.
4. If you want to provide a text description for the duplicate copy, type the description in the Description box.
5. Click the **Apply** button to create the duplicate folder.

Updating security folders

Use the following procedure to update a security folder:

1. On the Security Folders page, display the folder that you want to update.
2. Click the **Update** icon for the folder to display the Update Security Folder page.
3. If you want to update the folder name, the description, or both, make the desired updates in the Folder Name and Description boxes.
4. Do one of the following:
 - If you do not want to update user assignments for the folder, click the **Finish** button to update the folder.
 - If you want to update user assignments for the folder, click the **Continue** button to display the Step 2: Security Folder User Assignments page, on which you can make the desired updates to the user assignments.

When you have finished updating the user assignments, click the **Finish** button to update the folder.

Deleting security folders

Use the following procedure to delete a security folder:

1. On the Security Folders page, display the folder that you want to delete.
2. Click the **Delete** icon for the folder.
3. Click **Yes** in response to the prompt to delete the folder.

Customizing the Home Page

Through the Customize Homepage facility, you can customize the information that is displayed on the Home page, as follows:

- In the System Name section, you can specify the name of the system to which you are logged in. If you change the name, you must log out and log back in to see the updated name.

- In the Information section, you can specify both the heading and the content for any information that you would like to display on the Home tab for your user account. The content can be either unformatted text or in HTML format.
- In the Shortcuts section, you can define links to frequently used Web sites, and you can delete links that you no longer need. You can create headings to organize the links, and you can reorder the headings and links that you have specified.

To create a new link or heading, click the **Add new Heading or Link** button, then select **Heading** or **Link** in the Type box, as appropriate.

To delete a heading or link, click the **Delete** icon for the heading or link that you want to delete.

To reorder headings and links, do the following:

1. Click the **Order** button to display the Reorder Shortcuts page.
2. Select one or more headings or links that you want to reorder.
3. Use the arrow buttons to move the selected headings and links to the desired location.
4. Click the **Apply** button.

Working with Rule Approval Status

This chapter covers the following topics:

- About Rule Approval and Production Data Sets
- The Rule Approval Process
- The Rule Deletion Process

About Rule Approval and Production Data Sets

Enterprise Performance Foundation uses Oracle Approvals Management (AME) and Oracle Workflow to manage the approval status of rules.

Note: The approval process actually applies to versions of rules, as opposed to rules themselves. For simplicity's sake, however, the term "rule" is used in this chapter to refer to versions of rules.

A rule can write results to a production data set only if the status of the rule is Approved. Rules that have not been approved can write results only to non-production data sets. Note that a production data set is defined as one for which the appropriate attribute has been set on the data set member.

If you want to use production data sets, you must set up an approval hierarchy through Oracle Approvals Management (AME), where you must specify **FEM Approvals** as the transaction type. After the approval hierarchy has been defined, Enterprise Performance Foundation uses Oracle Workflow to route the approval requests to the appropriate approvers.

If a rule has been run and there are saved results that have been generated by the rule, the rule definition is locked, and you cannot change or delete the rule definition. You can only change or delete a rule definition if there are no existing results for that rule.

For further information about Oracle Approvals Management and Oracle Workflow, see the following:

- *Implementing Oracle Approvals Management*
- *Oracle Workflow User's Guide*

For more information about the processes for rule approval and deletion, see the following topics:

- The Rule Approval Process, page 11-2
- The Rule Deletion Process, page 11-2

The Rule Approval Process

When you first create a rule, the status for the rule is New. The rule must be approved before you can use it to write results to a production data set (you can, however, test an unapproved rule against a non-production data set).

When you submit a rule for approval, the status becomes Submit Approval. The approver can then either approve or reject the rule. Note that you cannot run a rule for which the status is Submit Approval against either production or non-production data sets.

If the approver approves the rule, the rule status changes to Approved, and you can now use the rule to write results to production data sets.

If the approver rejects the rule, the rule status changes to Not Approved. If there are no existing results for that rule definition, you can make changes to the rule definition and resubmit the rule for approval.

If you try to change the definition for an approved rule for which there are no existing results, a backup copy of the approved rule definition is created. The status of this backup copy of the definition is Not Approved. You can then revert to this backup copy of the rule definition, edit it as desired, and then submit the revised definition for approval.

The Rule Deletion Process

To delete a rule definition for which the status is Approved, you must obtain approval to delete the rule before it can be deleted. Therefore, to delete an approved rule, you must submit the rule definition for deletion approval.

Click the **Delete** icon for a rule on the rule's home page submit the rule definition for deletion approval. This action launches the approval process and routes the request for deletion to the appropriate approver.

When you submit a rule for deletion approval, the definition status for the rule becomes Submit Delete. Note that you cannot run a rule for which the status is Submit Delete against either production or non-production data sets.

If the approver approves the deletion of an approved rule, the rule is automatically deleted. If the approver rejects the deletion, the rule definition status is reset to Approved.

Rules with a status of New — that is, rules that have not been submitted through the approval process — do not require approval for deletion. As long as there are no results in a non-production data set as a result of running such a rule, you can simply delete the rule.

Implementing Enterprise Performance Foundation with Oracle General Ledger

This chapter covers the following topics:

- Introduction
- General Ledger Rules Overview
- Defining General Ledger Rules
- Synchronizing General Ledger and Enterprise Performance Foundation
- Miscellaneous Notes and Best Practices for Enterprise Performance Foundation and Oracle General Ledger Rules

Introduction

Oracle General Ledger is the central repository of financial information for a firm running the Oracle E-Business Suite. The information contained therein is critical for activities like financial and management reporting, profitability analysis, budgeting and planning, consolidation.

Several Corporate Performance Management applications enable these activities on Oracle General Ledger based data including:

- Oracle Enterprise Planning and Budgeting
- Oracle Financial Consolidation Hub
- Oracle Profitability Manager
- Oracle Transfer Pricing / Oracle Risk Manager

These applications leverage Enterprise Performance Foundation for analytical processing using Oracle General Ledger data as well as data from other sources. It is therefore necessary for the source Oracle General Ledger data to be first transferred and stored in this underlying architecture before any analysis or planning can be conducted.

Enterprise Performance Foundation includes a set of General Ledger Rules for you to specify what Oracle General Ledger data will be available for the Corporate Performance Management applications. This chapter provides you with all the information you need to set up and execute these rules.

General Ledger Rules Overview

General Ledger Rules define how Oracle General Ledger charts of accounts, calendars, and balances will be represented in Enterprise Performance Foundation. This overview section provides a high level description of the General Ledger Rules architecture.

The following subsections describe important constructs used in the setup.

Note: Detailed information on defining and executing these constructs follows this Overview section.

Oracle Ledger Assignments

Oracle Ledger Assignments allows you to select the set of ledgers whose data will be represented in Enterprise Performance Foundation for analytical processing. The selection of ledgers also dictates which charts of accounts need to be associated with Enterprise Performance Foundation dimensions.

Dimensions and Dimension Rules

Before Oracle General Ledger balances can be transferred into Enterprise Performance Foundation, all necessary dimensions and dimension values must first be defined by creating and executing Dimension Rules and Hierarchy Rules. These rules determine how critical Oracle General Ledger entities are represented in Enterprise Performance Foundation.

There are 2 primary types of dimensions in Enterprise Performance Foundation:

1. Account Based Dimensions: These relate to Oracle General Ledger charts of accounts and require value sets to store dimension values.
2. Non Account Based Dimensions: These relate to other key Oracle General Ledger entities outside of the charts of accounts.

Account Based Dimensions

Account-based dimensions are those related to one or more segments in the Oracle General Ledger charts of accounts. You need to indicate how individual chart of account segments will relate to these dimensions so that the appropriate dimension values can be generated and maintained from the segment values. Some Enterprise Performance Foundation dimensions can be related to multiple segments, resulting in dimension values that are based on a concatenation of account segment values.

The following account based dimensions are available in Enterprise Performance Foundation:

1. Company Cost Center Organization
2. Natural Account
3. Line Item
4. Product
5. Intercompany
6. Channel
7. Geography
8. Project

9. Customer
10. Task
11. Entity
12. User Defined Dimensions 1-10

Note: Not all dimensions appear in the user interface in all cases, as only the processing key dimensions are displayed there.

For a successful implementation, each of the above account based dimensions must be mapped to one or more chart of accounts segments or be assigned a default value through a Dimension Rule. As a result, every valid account code combination is mapped to a set of dimension values through Dimension Rules specific to that chart of accounts.

Each Dimension Rule is associated with one chart of accounts and one dimension and contains information on how members of the selected dimension will be generated. For example, the Dimension Rule for the Line Item dimension defines which segment/segments of the charts of accounts should be associated with the Line Item dimension.

Global Value Set Combination: Usually, dimension values for account based dimensions are stored in value sets within Enterprise Performance Foundation in a Global Value Set Combination (GVSC). The GVSC is a construct in Enterprise Performance Foundation similar to the chart of accounts. Each chart of accounts in a ledger is associated with and represented by a GVSC in Enterprise Performance Foundation. It specifies one value set for each of the account based dimensions and this value set assignment comes from the definition of the Dimension Rules for a particular chart of accounts. Ledgers in Oracle General Ledger that have the same chart of accounts will all have the same GVSC in Enterprise Performance Foundation.

Note: If two assigned ledgers share a common chart of accounts, they will share the same set of mappings.

As an example consider Ledger 1 with a Dimension Rule "XYZ" used for mapping the natural account segment to the Account dimension. If Ledger 2 in the same group of assigned ledgers uses the identical chart of accounts as Ledger 1, then the same Dimension Rule "XYZ" must be used while mapping the Account Dimension.

Non Account Based Dimensions

Non account-based dimensions include key Oracle entities outside of the chart of accounts like:

- Ledgers
- Budgets
- Encumbrance Types
- Calendar Periods

It is mandatory that these items also be defined in Enterprise Performance Foundation to facilitate the balance transfer. The setup is straightforward for these objects. Budgets and encumbrance types are automatically maintained when they are selected as part of the Balances transfer definition. Ledgers are setup when they are included through Oracle Ledger Assignments.

A single version of the calendar rule is seeded to facilitate the propagation of the Oracle General Ledger calendar in the corresponding Enterprise Performance Foundation calendars and calendar periods dimension. Users are not allowed to modify this calendar rule.

Hierarchies and Hierarchy Rules

Oracle General Ledger supports the creation of hierarchy structures for chart of accounts segment values. Since these hierarchy structures are widely used for analytic processing and management reporting, General Ledger Rules support creating and maintaining these hierarchies in Enterprise Performance Foundation as well. You can specify which chart of accounts hierarchies you wish to propagate and maintain in Enterprise Performance Foundation through Hierarchy Rules. Each Hierarchy Rule will result in a single top hierarchy in Enterprise Performance Foundation.

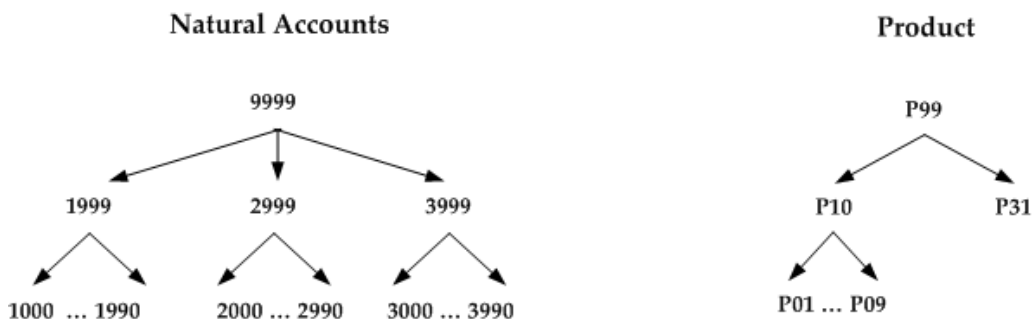
The chart of accounts segments and values required to build the hierarchy are based on the Dimension Rule definition associated with the Hierarchy Rule. In the case of dimensions linked to a single segment, the dimension hierarchy will be a copy of the hierarchy associated with the segment in Oracle General Ledger. You need to select one of the member parent values to be the top parent value of the dimension hierarchy. The program will then construct the hierarchy with information from the segment value set starting with the chosen top parent.

In the case of dimensions linked to multiple chart of accounts segments, the Hierarchy Rules definition requires that you specify a top parent for each segment selected in the Dimension Rule. With multi-segment hierarchies, you can specify a segment traversal order for creating the hierarchy that is different from the segment order used to construct the dimension members.

Example - Multi Segment Hierarchy Rule

Assume that a user defined Dimension Rule for "Dimension1" is based on the Natural Account and the Product segments. The rule specifies that members will be created by concatenating Natural Account values first, followed by Product segment values.

The following diagram displays the segment hierarchies for both these segments in Oracle General Ledger.



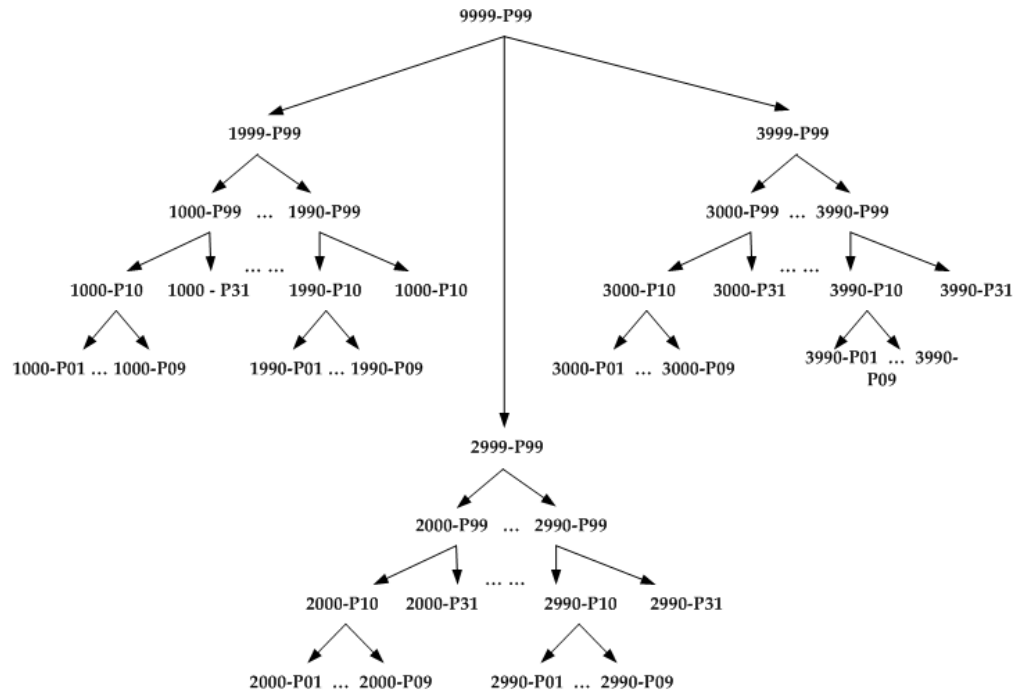
Scenario 1

The following table lists the details of the "Product Account Hierarchy" rule that can be defined for this dimension along with its contextual details:

Hierarchy Rule	Segment	Traversal Order	Starting Parent Value
Product-Account Hierarchy	Natural Account	Second	9999
Product-Account Hierarchy	Product	First	P99

Based on the above rule, the following diagram shows the resulting hierarchy in Enterprise Performance Foundation.

Product-Account Hierarchy



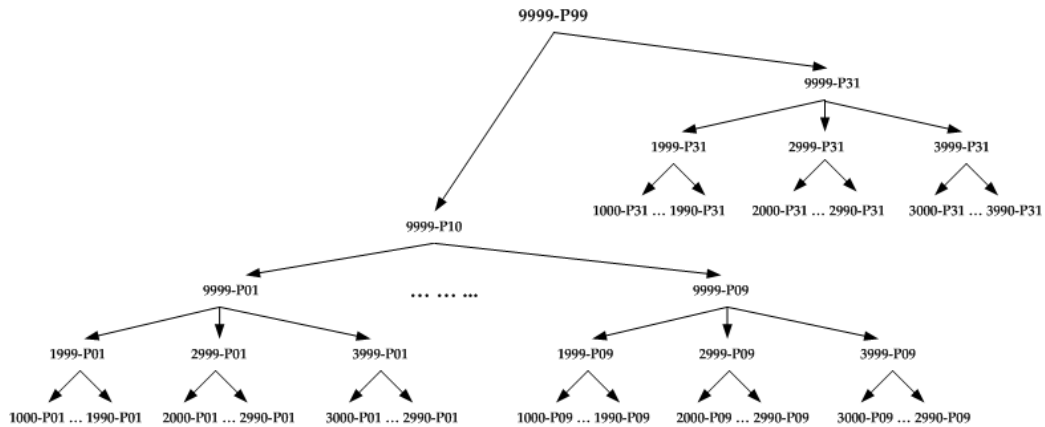
Scenario 2

The following table lists the details of a new Hierarchy Rule "Account Product Hierarchy" that can be defined for this dimension along with its contextual details:

Hierarchy Rule	Segment	Traversal Order	Starting Parent Value
Account-Product Hierarchy	Natural Account	First	9999
Account-Product Hierarchy	Product	Second	P99

Note that the traversal order for this rule is different than in Scenario 1. Based on the above rule, the following diagram shows the resulting hierarchy in Enterprise Performance Foundation for this dimension's members.

Account-Product Hierarchy



In both scenarios above, note that the parent dimension members are still created using the segment order specified in the Dimension Rule (i.e. Natural Account - Product). However, the parent child relationships are created based on the traversal order specified in the Hierarchy Rules. As a result, the rollup structure and the parent values of the hierarchies are different.

The concatenated hierarchy is built from the component hierarchies using a "bottom up" approach. This means that when the program is building the Product-Account Hierarchy, it starts with the bottom level of the Product hierarchy and the Natural Account hierarchy (i.e. the detail children). It then traverses up the Product Hierarchy first (while Natural Account stays at the bottom level) until it reaches the top parent (i.e. P99). After that, it switches to traverse up the Natural Account hierarchy until it reaches 9999. Therefore, when you look at the resulting hierarchy from the top down, the Product hierarchy will actually appear to be the last segment traversed

The parent value "T", if used, has no special meaning and is treated exactly like any other parent value i.e. it has to have child assignments in the segment hierarchies to be used as a top parent.

Note: This is slightly different from how the Oracle General Ledger uses the "T" segment value. Oracle General Ledger assumes that "T" is the parent of all detail members in the segment.

Hierarchy Rule Versioning

With the passage of time, the Oracle General Ledger segment hierarchies may change or you may desire to maintain the hierarchy using different criteria. By creating Hierarchy Rule versions, General Ledger Rules allows you to update hierarchies in Enterprise Performance Foundation while maintaining prior hierarchy structures for comparison.

A new Hierarchy Rule version is created by end dating the prior version. Each version represents a particular hierarchy structure i.e. a specific top parent (the starting point of the hierarchy) with a particular set of parent child relationships.

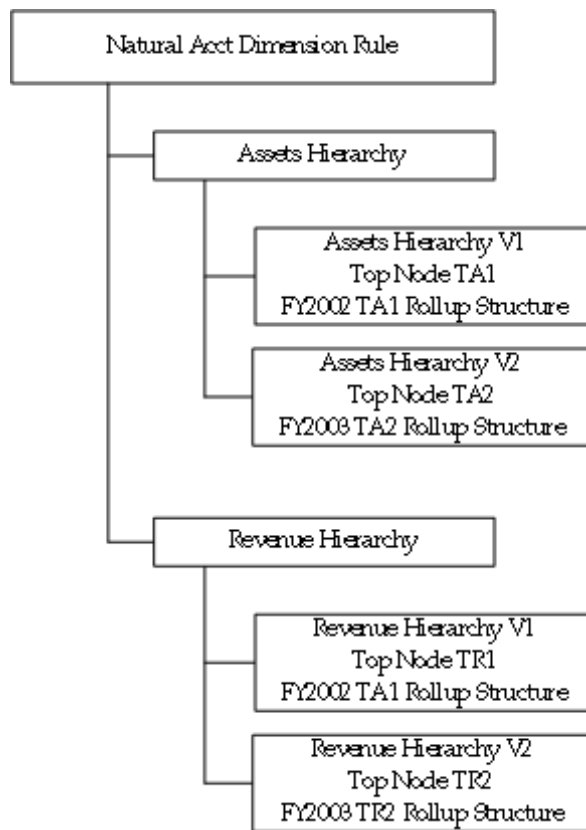
Note: The Hierarchy Rule itself is simply a collection and pointer to one or more versions of the rule. It does not itself furnish the hierarchy structure. The version of the hierarchy is where the parent child relationships and top level parents are stored.

Here are some common cases where you may want to create a new version of a particular Hierarchy Rule:

- There are changes in the parent-child relationships in the hierarchy, and the current hierarchy structure has been used by other Corporate Performance Management applications for analysis.
- There is a change in the top parent that represents the hierarchy structure.
- You want to preserve a copy of the current hierarchy structure.

Otherwise, when you execute a Hierarchy Rule version, the process will always replace the current structure with the latest from Oracle General Ledger, including addition of new children values, or movement of children from one parent to another.

The following diagram provides an example of the hierarchy rule versions discussed above:



As shown in the diagram above, each Hierarchy Rule is based on a dimension and defines a hierarchy. You can create multiple hierarchies in Enterprise Performance Foundation, like an Assets hierarchy, Revenue hierarchy etc. replicating those that are associated with the related segment in Oracle General Ledger. For each Hierarchy Rule, multiple versions can be created, with each version representing a particular hierarchy structure.

Balances Rules

Balances Rules define how and which balances in a particular ledger will be represented in Enterprise Performance Foundation. Only ledgers assigned to the implementation are available for use in Balances Rules.

One Balances Rule is required for each balance type (Actuals, Budget, or Encumbrance) within this ledger. For a particular balance type, each Balances Rule also has its own set of options and settings for controlling the balances transfer. You can use these to transfer a subset of data. For example, for the ABC Ledger, a Balances Rule can be set up to transfer Actual balances for Jan-Mar 2004 and for Companies 1-5 only.

Balances transfer options and settings include the following:

- Budget or encumbrance type, if applicable.
- Balancing segment values.
- Currency type. For example "Cumulative Functional" only, or "Entered with Functional Equivalent" balances.
- Translated balances

Run the Balances Rules to make Oracle General Ledger balances available in Enterprise Performance Foundation. Balances data is extracted from the GL_BALANCES and GL_DAILY_BALANCES tables of selected ledgers in Oracle General Ledger and the balances loaded into FEM_BALANCES.

The execution mode for these rules can be either "Incremental Update " or "Snapshot Only." In the former case, balances data is initially brought in as a complete snapshot and subsequently maintained incrementally. In the latter case, balances data is always maintained by taking a new snapshot of the Oracle General Ledger balances.

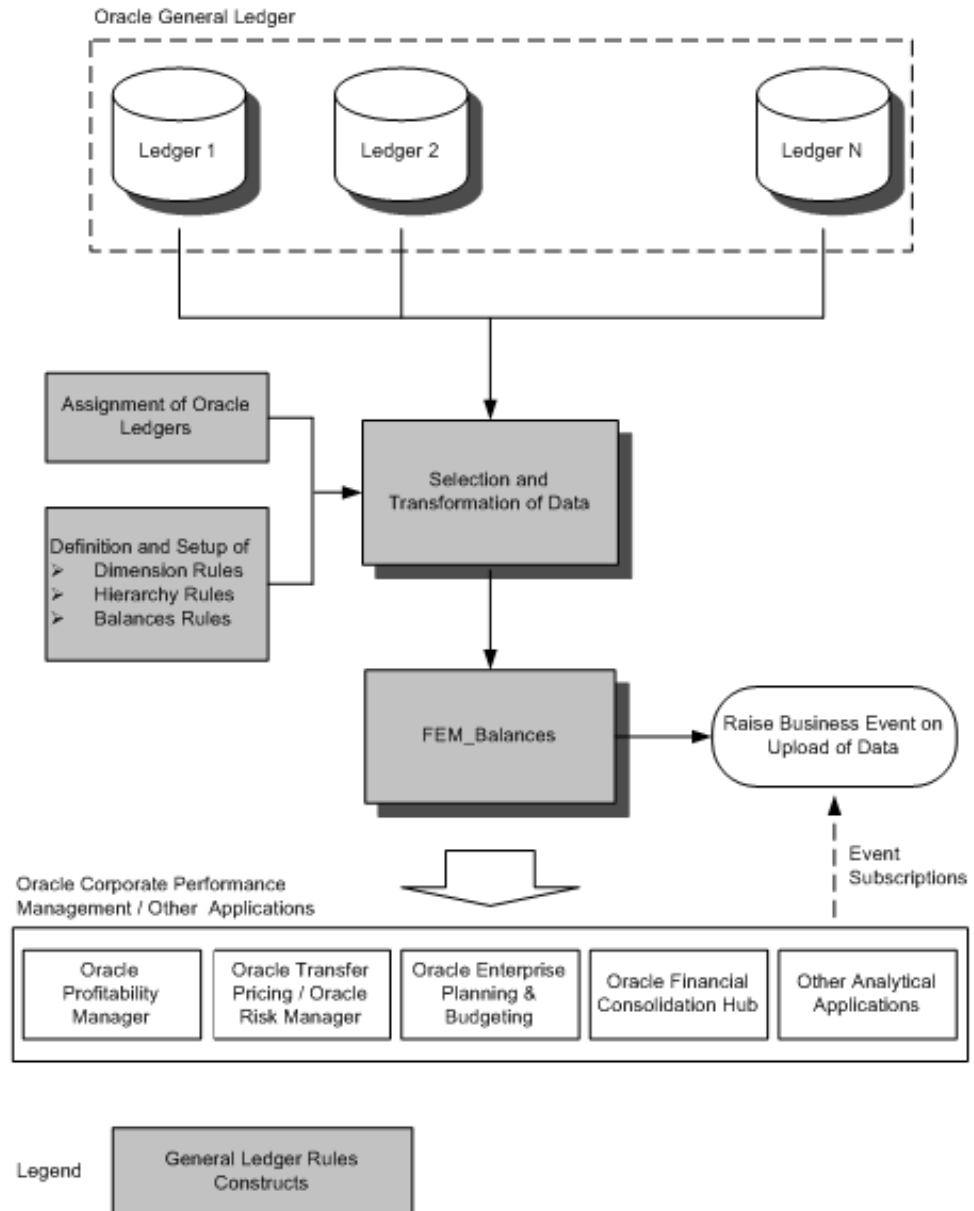
Balance Rules can be versioned by effective date and this determines the accounting period data to which the rule applies.

Business Event Publishing

When Balances Rules are run and new data is brought into Enterprise Performance Foundation, business event messages are published. This information can be used by Corporate Performance Management applications to determine whether they should uptake any of the data changes. The raising of business events hence provides flexibility in the processing of changes in balances.

General Ledger Rules Architecture

The following diagram provides an overview of the process and how the General Ledger Rules work together.



Perform the following steps to complete an initial implementation of the General Ledger Rules:

1. Include the relevant ledgers into the scope of the implementation through Oracle Ledger Assignments. The data from these ledgers will be pulled into the Enterprise Performance Foundation framework for analysis and reporting.
2. For each chart of accounts used by the assigned ledgers, all relevant Dimension Rules will automatically be created for you. You need to update the option on how members of each dimension will be generated in Enterprise Performance Foundation based on the chart of accounts segments.
3. (Optional step) For each relevant dimension, specify Hierarchy Rules for associated hierarchies to be represented in Enterprise Performance Foundation.

4. For each assigned ledger, create Balances Rules that define the data that is to be transferred to Enterprise Performance Foundation. Each rule contains specific information on what balances data will be extracted from Oracle General Ledger when the rule is executed.
5. To create and maintain dimension members, hierarchies, and Oracle General Ledger balances in the Enterprise Performance Foundation framework, select the appropriate rules and run them.
6. If applicable, schedule rule executions using the standard concurrent manager scheduling feature.
7. Periodically update Enterprise Performance Foundation with applicable changes from Oracle General Ledger for set up, definitions and balances data.

Note: The necessary non-account based dimension objects like ledgers, calendars, budgets, and encumbrance types are propagated to Enterprise Performance Foundation during various phases of setup and execution of the General Ledger Rules.

For ongoing maintenance, you will need to continue running the General Ledger Rules to ensure that the latest firm data is always represented in Enterprise Performance Foundation.

When new Oracle General Ledger balances are transferred into Enterprise Performance Foundation, Corporate Performance Management applications are notified through a published business event because the new and changed data might invalidate results based on prior balances.

Defining General Ledger Rules

The following sections provide detailed information on setting up General Ledger Rules constructs introduced earlier.

Prerequisite

Before General Ledger Rules and constructs can be set up, the Enterprise Performance Foundation setup must first be completed.

Note: For more information, refer to appropriate sections of the *Enterprise Performance Foundation User's Guide*.

1. Setup Oracle Ledger Assignments

Oracle Ledger Assignments is the group of Oracle General Ledger ledgers whose data is represented in Enterprise Performance Foundation.

Note: For an introduction to Ledger Assignments, refer to the General Ledger Rules Overview, page 12-2 section earlier in this chapter.

Topic	Navigation Path
Oracle Ledger Assignments	Using the Enterprise Performance Foundation Administrator (or equivalent) responsibility, navigate to the Data Rules tab and then the Oracle Ledger Assignment subtab [Ledger Assignments Window]. Assign or unassign ledgers as appropriate.

Ledger Assignments

Assignments to the Source Ledger Group represent the collection of ledgers from which you report and analyze financial information. Revert Save

Search

Search Go

Assign Ledgers

Ledger	Description	Chart of Accounts	Unassign
Progress Canada	Vision Progress Canada Set of Books	Progress Canada Accounting Flex	

There is no limit to the number or type of ledgers that can be included using Oracle Ledger Assignments; assign all ledgers that you intend to analyze through Enterprise Performance Foundation at any given point in time. The ledgers can be on different charts of accounts, calendars, and/or functional currencies.

A ledger can be unassigned at any time, even if Balances Rules have been defined and executed for it. However you cannot execute Balances Rules for ledgers not currently displayed as assigned under Ledger Assignments.

2. Setup Dimension Rules

Dimension Rules detail how key accounting entities in Oracle General Ledger relate to Enterprise Performance Foundation dimensions.

Note: For a detailed introduction to Dimension Rules, refer to General Ledger Rules Overview, page 12-2 earlier in the chapter.

As noted earlier, there are 2 primary types of dimensions in Enterprise Performance Foundation:

1. Account Based Dimensions: These relate to Oracle General Ledger charts of accounts
2. Non Account Based Dimensions: These relate to other Oracle General Ledger entities like calendars

Setup of Account Based Dimension Rules

Before balances can be transferred from Oracle General Ledger to Enterprise Performance Foundation, all dimensions with a Dimension Rule generated must be mapped to one or more chart of accounts segments or be assigned a default value. This mapping definition between segments of a chart of accounts and the account based dimension represents a Dimension Rule.

Note that default Dimension Rules are automatically generated for all value set required dimensions using a default mapping mechanism. You can modify these rules as required.

Topic	Navigation Path
Create Dimension Rules	Using the Enterprise Performance Foundation Administrator (or equivalent) responsibility, navigate to the Data Rules tab and then the Oracle Dimension Sources subtab.

The screenshot shows the Oracle Enterprise Performance Foundation Data Rules interface. The page title is "Account Based Dimension Rules" for the "Chart of Accounts" under "Progress Canada Accounting Flex". The last run date is 21-Jun-2005 and the last run by is FEMINTG. The interface includes a "Submit All Rules" button and a table of dimension rules.

Dimension	Mapping	View	Update	Define Hierarchy Rules	Undo
Company Cost Center Organization	Copy: Entity				
Line Item	Copy: Account				
Channel	Assign Single Value: Default				
Customer	Assign Single Value: Default				
Natural Account	Copy: Account				
Product	Assign Single Value: Default				
Task	Assign Single Value: Default				

While the Dimension column displays the list of processing key dimensions in the FEM_BALANCES table, the Mapping column shows the dimension's current mapping definition stored in the related Dimension Rule. For the given chart of accounts, the mapping definition in each rule determines how members of the associated dimension will be generated.

There are three types of mapping options available:

1. Copy Values from Single Segment - copy all segment values from the selected segment to the dimension. In this case, the dimension will be directly related with that segment.
2. Concatenate Values from Multiple Segments - dimension values will be a concatenation of multiple chart of account segments, separated by a dash. In this case, the concatenation sequence must also be specified.
3. Assign Single Value - the dimension will not be mapped to any specific chart of accounts segment value. Instead, a specific value is assigned to the dimension.

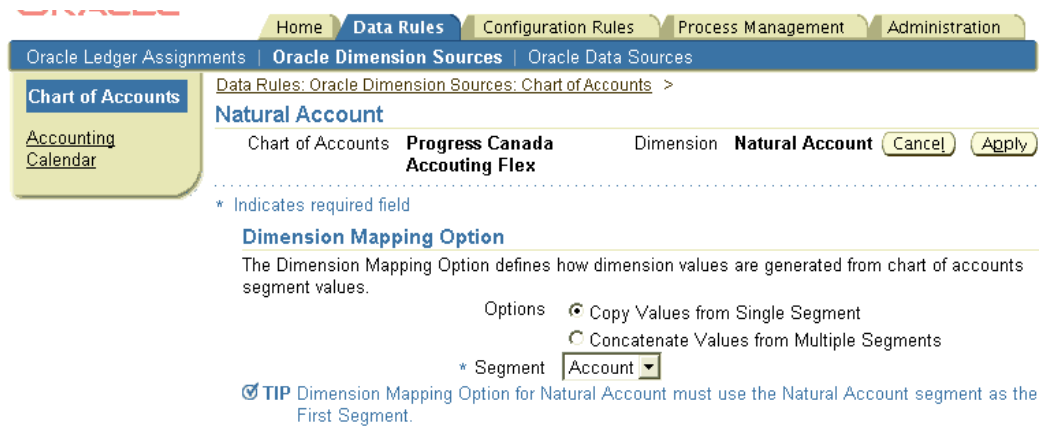
Default Mapping Mechanism

The default mapping mechanism for value set required dimensions is as follows:

Dimension	Default Mapping
Company Cost Center Organization	Copy Values from the Balancing Segment.
Line Item	Copy Values from the Natural Account Segment.
Natural Account	Copy Values from the Natural Account Segment.
All other active account based dimensions	Assign a Default Single Value Displayed as "Assign Single Value: Default"

Note: The Balancing Segment and the Natural Account segment are identified by their respective segment qualifiers in the chart of accounts.

Click the View or Update icons for more details on a particular Dimension Rule.



Mapping of Accounts Based Dimensions

Dimension	Potential Chart of Accounts Segments	Mapping Options Available	Other Rules
Company Cost Center Organization	Company, Cost Center, Company + Cost Center	Copy values from Single Segment Concatenate Values from Multiple Segments	Default option is Copy Values from Single Segment, with both the Balancing and Cost Center segments as candidates. If Concatenate Values from Multiple Segments is chosen, then two segments are allowed. The First Segment must be the Balancing segment and the Second Segment must be the Cost Center segment.

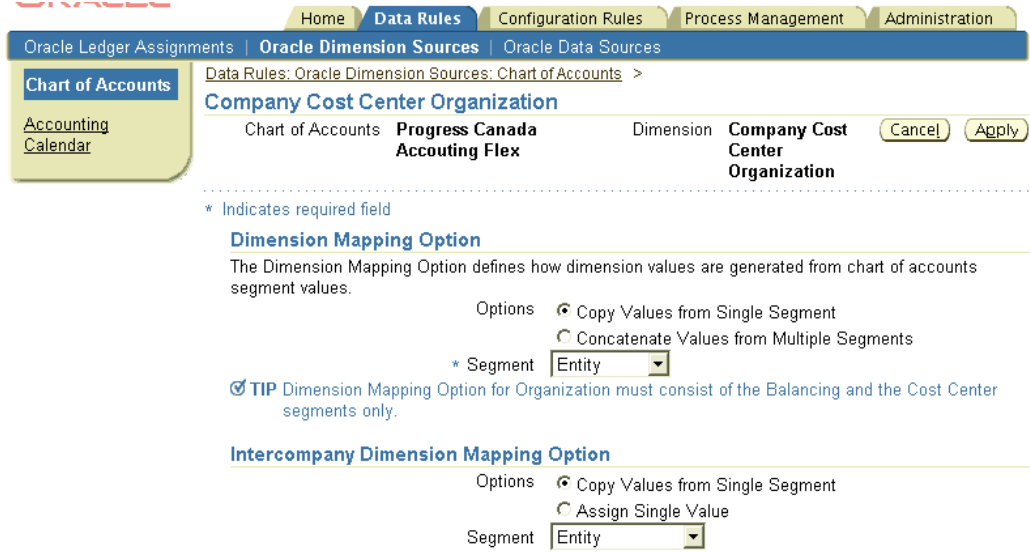
Dimension	Potential Chart of Accounts Segments	Mapping Options Available	Other Rules
Line Item	Natural Account and up to four additional segments	Copy values from Single Segment Concatenate Values from Multiple Segments	Default option is Copy Values from Single Segment with the Natural Account segment defaulted. You cannot change this default value to any other segment. If Concatenate Values from Multiple Segments is chosen, one of the segments must be the Natural Account segment.
Natural Account	Natural Account and one additional segment	Copy values from Single Segment Concatenate Values from Multiple Segments	Default option is Copy Values from Single Segment with the Natural Account segment defaulted. You cannot change this default value to any other segment. If Concatenate Values from Multiple Segments is chosen, the First segment defaults to the Natural Account segment and it cannot be changed.
Entity	Balancing segment	Copy values from Single Segment Assign Single Value	Default option is Copy Values from Single Segment with the Balancing segment defaulted. This Balancing Segment selection cannot be changed.
Intercompany	Up to two segments	Copy values from Single Segment Concatenate Values from Multiple Segments Assign Single Value	Refer to the section "Mapping the Intercompany Dimension" below.
Channel Geography Customer Product Project Task User Defined 1-10	All segments	Copy values from Single Segment Concatenate Values from Multiple Segments Assign Single Value	Default option is "Assign Single Value Default." This is valid when the dimension has no chart of accounts counterpart.

Mapping the Intercompany Dimension

Intercompany Dimension Rules need to follow the mapping of the Company Cost Center Organization dimension due to the close relationship between the two dimensions. Hence, you need to run the Dimension Rule for the Company Cost Center Organization dimension before selecting an appropriate mapping for the Intercompany dimension.

There are two different cases based on the mapping of the Company Cost Center Organization Dimension:

1. Company Cost Center Organization Dimension Option is Copy Values from Single Segment
2. Company Cost Center Organization Dimension Option is Concatenate Values from Multiple Segments



a. Company Cost Center Organization Dimension Option: Copy Values from Single Segment

Company Cost Center Organization Dimension Mapping Choice	Intercompany Dimension Mapping Option	Segment(s) Mapped to Intercompany Dimension
Balancing Segment	Copy values from Single Segment	Default option is Copy Values from Single Segment with the Intercompany segment defaulted (identified by the Intercompany qualifier). This value cannot be changed to any other segment. If the Intercompany segment is not identified, then all segments appear and you must choose the segment being used as the Intercompany segment.
Cost Center Segment	Copy values from Single Segment	Choose one of the segments as the "Inter Cost Center" segment.
Either Balancing or Cost Center Segment	Assign a Single Value	Value is automatically "Default." Use this option if Intercompany / Intercost center balancing is not used in the implementation.

b. Company Cost Center Organization Dimension Option: Concatenate Values from Multiple Segments

Company Cost Center Organization Dimension Segment	Intercompany Dimension Mapping Option	Segment(s) Mapped to Intercompany Dimension
The first Segment defaults to the Balancing segment and the Second Segment to the Cost Center segment.	Concatenate Values from Multiple Segments	The First Segment defaults to the Intercompany segment (identified by the Intercompany qualifier). If the Intercompany segment is not identified, then all segments appear and you must choose the segment being used as the Intercompany segment. For the Second Segment, choose one of the segments as the "Inter Cost Center" segment.
The first Segment defaults to the Balancing segment and the Second Segment to the Cost Center segment.	Assign a Single Value	Value is automatically "Default." Use this option if Intercompany / Intercost center balancing is not used in the implementation.

Setup of Non Account Based Dimension Rules

In addition to the dimensions described in previous sections, the following non - account based dimensions are required to support the transfer process:

- Ledgers
- Calendars
- Budgets
- Encumbrance Types

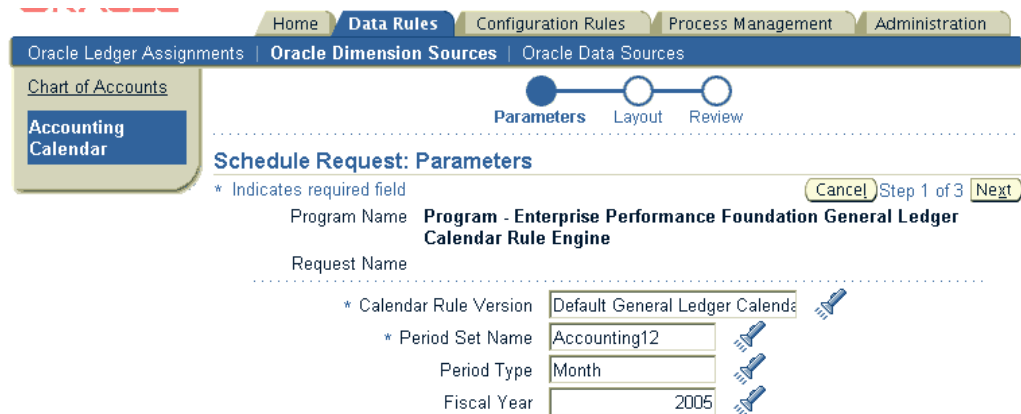
Ledgers

The propagation of ledgers from Oracle General Ledger to Enterprise Performance Foundation is transparent to the user and is automatically conducted when a ledger is selected through Oracle Ledger Assignments.

Calendars

Oracle General Ledger calendar definitions are transferred to Enterprise Performance Foundation by submitting the seeded Calendar Rule. Each submission will allow you to bring over period definitions of one fiscal year into Enterprise Performance Foundation. If you add new fiscal years in the Oracle General Ledger calendar, then you must resubmit the Calendar Rule to synchronize them.

Topic	Navigation Path
Setup the Calendar dimension in Enterprise Performance Foundation	Using the Enterprise Performance Foundation Administrator (or equivalent) responsibility, navigate to the Data Rules tab and then the Oracle Dimension Sources subtab. Drill down into the Accounting Calendar.



The only calendar rule version available is the "Default General Ledger Calendar Rule Version." The Calendar Rule execution transfers calendar information and generates the appropriate period hierarchy structures - periods, quarters, etc. in the Enterprise Performance Foundation Calendar Period dimension. You must run the Calendar Rule for any Fiscal Year whose balances will be transferred to Enterprise Performance Foundation, or whose periods will be used for budgeting purposes.

Encumbrance Types and Budgets

Definition of encumbrance types and budgets in Enterprise Performance Foundation is also transparent to users. It is automatically handled when a particular budget or encumbrance type is selected when defining a Balances Rule.

3. Setup Hierarchy Rules

For each relevant dimension, you can specify Hierarchy Rules for associated hierarchies to be represented in Enterprise Performance Foundation.

Note: For a detailed introduction to Hierarchy Rules, refer to the General Ledger Rules Overview, page 12-2 section earlier in the chapter.

Topic	Navigation Path
Create Hierarchy Rules	Using the Enterprise Performance Foundation Administrator (or equivalent) responsibility, navigate to the Data Rules tab and then the Oracle Dimension Sources subtab. For the relevant dimension, click the associated Hierarchy Rules icon to create Hierarchy Rules.

If the associated dimension is based on a single segment, then enter the single Starting Parent Value for the hierarchy. In the multi segment case, enter the Starting Parent Value for each segment along with the Parent Traversal Order.

ORACLE

Home | Data Rules | Configuration Rules | Process Management | Administration

Oracle Ledger Assignments | Oracle Dimension Sources | Oracle Data Sources

Data Rules: Oracle Dimension Sources: Chart of Accounts > Account Based Dimension Rules > Hierarchy Rules >

Create Hierarchy Rule

Chart of Accounts: **COA1** | Dimension: **Natural Account** | Version: **Default Version** | Effective Start Date: **01-Jan-1900** | Effective End Date: **01-Jan-2500** |

* Indicates required field

* Rule Description: [Text Field]

Generate Hierarchy in Standard and Flattened Format

Create as Sequence Enforced Hierarchy

TIP A corresponding hierarchy will be created and named after the rule. You can use Dimension Hierarchy Manager to change the hierarchy name once it is created.

Version

* Version Description: [Text Field]

* Start Date: 01-Jan-1900 (example: 14-Dec-2005)

* End Date: 01-Jan-2500

Hierarchy Traversal Sequence

The hierarchy traversal sequence defines the order of segments whose hierarchies will be traversed to construct the concatenated hierarchy.

Segment	Parent Traversal Order	*Starting Parent Value
Account	First	[Text Field]
Department	Second	[Text Field]

The following table provides further information on select fields in the Create Hierarchy Rule window.

Field	Description
Generate Hierarchy in Standard and Flattened Format	Whether to automatically run the program "DHM:Flatten Hierarchy Version" to generate parent child relationships in Enterprise Performance Foundation between a parent and all its descendents.
Create as Sequence Enforced Hierarchy	When this option is enabled, levels will automatically be created and assigned for the dimension members, and the hierarchy will be skip level sequence enforced. Otherwise, no levels will be created, and the hierarchy will not be sequence enforced.
Starting Parent Value	<p>A Hierarchy Rule version requires the designation of a top parent for the hierarchy.</p> <p>This top parent must be a parent segment value. The entire hierarchy structure from this point down to the lowest child level values in Oracle General Ledger is replicated and maintained as a dimension hierarchy in Enterprise Performance Foundation.</p> <p>A dimension with concatenated segments will require a Starting Parent for each segment.</p>
Start / End Dates	<p>To prevent the prior hierarchy version from being overridden, you can end date a hierarchy rule version and thereby preserve the corresponding hierarchy structure for analysis.</p> <p>Each Hierarchy Rule can have multiple versions, but the effective dates for each version cannot overlap.</p> <p>When a new version of the hierarchy is created, its name will follow the convention "<Hierarchy Rule Version Name>: <effective start date>."</p>

4. Setup Balances Rules

Balances Rules define how balances of the assigned Oracle General Ledger ledgers will be represented in Enterprise Performance Foundation.

Note: For an introduction to Balances Rules, refer to the General Ledger Rules Overview, page 12-2 section earlier in this chapter.

Each Balances Rule is created for a particular ledger and balance type and is setup in four stages.

Topic	Navigation Path
Setup Balances Rules	Using the Enterprise Performance Foundation Administrator (or equivalent) responsibility, navigate to the Data Rules tab and then the Oracle Data Sources subtab [Balances Rules Window] to create, update, or delete Balances Rules.

Stage 1: Rule Definition

The rule definition captures the ledger and balance type (Actual, Budget, Encumbrance) to be processed in the execution of the rule.

Based on the balance type selected, additional parameters must be selected as options (See Stage 4: Balances options below).

By selecting the Maintain Quarterly Balances checkbox, you can opt to calculate the QTD balances of each period as part of the transfer process.

Note: QTD balances are currently not used by Corporate Performance Management products, but only tracked for reporting purposes. You may want to weigh the cost benefit advantage of making this selection as the calculation will require additional processing resources.

Stage 2: Version Details

Three different parameters must be entered:

1. Start Date / End Date
2. Output Dataset
3. Execution Mode (Incremental Update vs. Snapshot Only)

Home | **Data Rules** | Configuration Rules | Process Management | Administration

Oracle Ledger Assignments | Oracle Dimension Sources | **Oracle Data Sources**

GL Balances

Define Rule | **Add Version** | Select Values | Define Options | Review

Create Balances Rule: Add Version

Cancel | Back | Step 2 of 5 | Next

Rule **Test Rule 1** | Ledger **Progress Canada**

* Indicates required field

* Version: Version 1

Description: [Empty text area]

* Start Date: 01.01.2005 (example: 07.17.2005)

* End Date: 08.01.2005

* Output Dataset: Default

Execution Mode: Incremental Update Snapshot Only

Start Date / End Date:

Balances Rules can change over time. For example, for a given Balances Rule, you may want to make changes in balancing segment values, the translated currencies involved, budget and encumbrance type assignments, among others.

Enterprise Performance Foundation allows you to create multiple versions of a Balances rule, each with non overlapping date ranges. Note that both start and end date of a version can be defaulted from the FEM: Default Effective Start Date and FEM: Default Effective End Date profile option settings.

Versioning provides the ability to preserve an audit trail of the data transfer. Once a Balances Rule version has been executed to generate data in Enterprise Performance Foundation, it can no longer be modified as that version is archived to provide the audit trail. However, you can end date the version and create a new one.

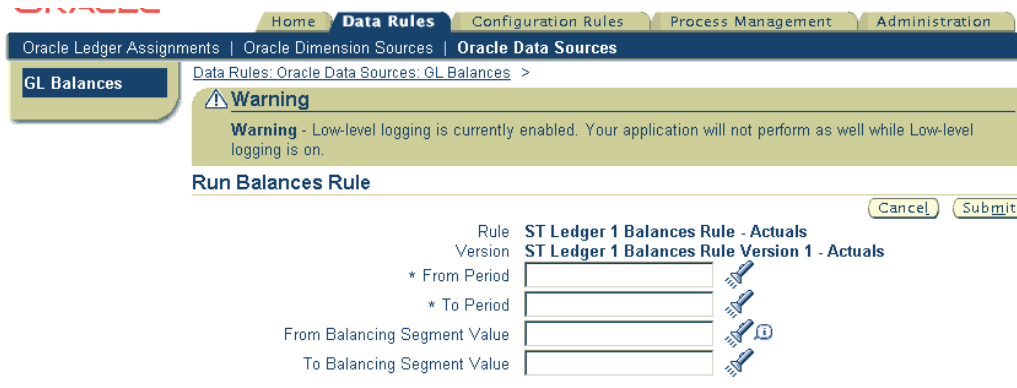
Note: If for any reason you want to modify a locked version of the Balances Rule, you need to reverse all balances that have been loaded using the rule version.

For more information, refer to Reversal of Executed Rules later in this section.

Output Dataset

You only need to specify an output dataset for a Balances Rule of the Actual balance type. For the other balance types, the output dataset is generated for you.

When you run a Balances Rule, you execute a particular version of the rule for a specific period range. This period range is entered at execution time.



To preserve the integrity of the data, one and only one Balances Rule version is applicable to the data for a given accounting period within the specified period range. The following points are critical in this regard:

- It is your responsibility to ensure that at least one Balances Rule version is associated with the data of every accounting period in the period range for which the rule is run. Hence, when defining versions, remember to maintain continuity between accounting periods such that all relevant data is captured in the period range.

Note: Also refer to Running Balances Transfer in Gapless and Sequential Period Order.

- Since only one Balances Rule version can be applicable to the data of a given accounting period, multiple versions of the rule must have non overlapping dates (enforced by the General Ledger Rules architecture). The end date of a rule version must be after the end date of the last period for which balances have been transferred to Enterprise Performance Foundation using that version.
- For a specified period range, Balances Rules utilize the version that is active on the end date of each accounting period within the date range.

As an example of this last point, consider a Balances Rule ABC with two versions V1 and V2. Version V1 has its start/end dates as Jan 1 - Mar 15. Version V2 has its start/end dates as Mar 16 - Dec 31. Assume also that the Oracle General Ledger calendar is a monthly calendar with starting/ending dates as the beginning and ending of the month.

If V1 of rule ABC is executed for the Jan - Feb date range, it will transfer data successfully for these two accounting periods. If V1 is executed for the Jan - Mar date range, it will fail as this version is not active on the March accounting period's end date.

Balances Output Dataset: In order to maintain data integrity in Enterprise Performance Foundation, balances that have been transferred from Oracle General Ledger must be represented as a single source of truth. Hence, for each combination of

- Ledger
- Period
- Balance type (actual, budget or encumbrance type)

as applicable, only one copy of Oracle General Ledger balance data is allowed in Enterprise Performance Foundation. The architecture uses the concept of a "dataset" to

support and validate this rule. Balances for each period can only be transferred into one dataset by a particular version of a Balances Rule and cannot be duplicated and stored in any other datasets. This ensures integrity in the base data that will be used for reporting, modeling, and analysis in Enterprise Performance Foundation.

In the case of a rule using the Actual balance type, you can name the output data set as appropriate. The system automatically creates and names the necessary target output data sets for Balances Rules using budgets and encumbrance types. Default names are based on the ledger and the budget / encumbrance type for which the output data set is created. You can use the dimension management tools in Enterprise Performance Foundation to update these names.

Execution Mode: Balances Rules provides two execution modes giving you more flexibility in when to transfer balances to Enterprise Performance Foundation.

"Snapshot Only" mode allows you to transfer Oracle General Ledger balances to Enterprise Performance Foundation only once per period by taking a copy of the balances at transfer time. Using this mode, existing snapshot data for a particular period must be reversed before you can load a new snapshot into the same period.

You can "undo" the appropriate rule and rerun it for the period(s) if you use Snapshot Only mode.

Note: For more information, refer to Reversal of Executed Rules, page 12-24 later in this section.

"Incremental Update" mode allows you to transfer Oracle General Ledger balances to Enterprise Performance Foundation multiple times in any period. The first execution of a Balances Rule with this mode takes a snapshot of the balances into Enterprise Performance Foundation. However subsequent runs just takes the changes in the balance since the last transfer. Incremental amounts are thus added to existing balances in Enterprise Performance Foundation.

Stage 3: Balancing Segment Values (BSV) Filter

In this stage of the setup, you can select the balancing segment values for which you wish to transfer balances.

Oracle Ledger Assignments | Oracle Dimension Sources | **Oracle Data Sources**

GL Balances

Define Rule | Add Version | **Select Values** | Define Options | Review

Create Balances Rule: Select Balancing Segment Values

Cancel | Back | Step 3 of 5 | Next

Rule **Test Rule 1**
Version **Version 1**
Ledger **Progress Canada**

Include Balancing Segment Values All Selected

Add Values

Entity	Description	Remove
03	City	
02	Provincial Government Ministry	
01	Federal Government	

Note that in addition to the segment values selected in the Balances Rule definition itself, you can also refine this selection at the time of executing the Balances Rules.

Only the list of balancing segment values selected as part of the Balances Rule definition will be applied when the rule is being submitted for the first time for the given period. The process will ignore the range of balancing segment values provided at submission time to ensure Enterprise Performance Foundation balances are initialized properly for the period. Subsequent submissions of the same rule for the same period will result in only the selected BSV's being updated.

Stage 4: Balances Options

For **Encumbrance Type Balances Rules**, you can select particular encumbrance types to be included in the balances transfer.

For **Budget Type Balances Rules**, you can limit the balances to individual budgets. You can then include translated balances for all or selected currencies associated with these budgets. Only if translated balances exist in the specific currency for the budget will such balances be transferred to Enterprise Performance Foundation.

For **Actuals Type Balances Rules**, you must select the type of balances to be included as:

- Cumulative functional currency balances only
- Both the entered and functional currency equivalent balances only, or
- Translated balances only. In this case you can choose "All" translated balances that exist or restrict the balance transfer to specific translated currencies.

The screenshot displays the Oracle Enterprise Performance Foundation user interface. At the top, there is a navigation bar with tabs for Home, Data Rules, Configuration Rules, Process Management, and Administration. Below this is a breadcrumb trail: Oracle Ledger Assignments | Oracle Dimension Sources | Oracle Data Sources. A sidebar on the left shows 'GL Balances' selected. A progress indicator shows five steps: Define Rule, Add Version, Select Values, Define Options (current step), and Review. The main content area is titled 'Create Balances Rule: Define Actual Balances Options'. It includes a 'Cancel' button, a 'Back' button, a dropdown menu showing 'Step 4 of 5', and a 'Next' button. Below these are fields for 'Rule' (Test Rule 1) and 'Version' (Version 1), and 'Ledger' (Progress Canada). The 'Currency Type' section has three radio button options: 'Cumulative Functional' (selected), 'Entered and Functional Equivalent', and 'Translated Only'. The 'Include Translated Balances' section has three radio button options: 'None' (selected), 'All Translated Currencies', and 'Selected Translated Currencies'.

Note that the Balances Rules transfer process does not perform any currency conversion. It assumes that all balances are prepared by the source Oracle General Ledger system.

Reversal of Executed Rules

You can reverse the creation of all data generated by a Balances Rule through the Executed Rule user interface.

Topic	Navigation Path
Reverse Executed Balances Rules	Using the Enterprise Performance Foundation Administrator (or equivalent) responsibility, navigate to the Process Management tab and then the Executed Rules subtab. All executed rules which affect the FEM_BALANCES table are displayed.

Select any of the executed rules to undo the resulting balances transfer to the FEM_BALANCES table.

Synchronizing General Ledger and Enterprise Performance Foundation

Once the setup of General Ledger Rules is complete, you can execute the rules to synchronize relevant data between Oracle General Ledger and Enterprise Performance Foundation.

Note: Refer to the section General Ledger Rules Architecture, page 12-8 for a high level overview of the synchronization steps.

Execute Dimension Rules

For the single segment mapping relationship, the process copies segment values from Oracle General Ledger to Enterprise Performance Foundation. For the multiple segments mapping relationship, the process concatenates the segment values in the user specified sequence to produce concatenated dimension values.

Topic	Navigation Path
Execute Dimension Rules	Using the Enterprise Performance Foundation Administrator (or equivalent) responsibility, navigate to the Data Rules tab and then the Oracle Dimension Sources subtab. Click the Submit All Rules button.
Undo Dimension Rule	Using the Enterprise Performance Foundation Administrator (or equivalent) responsibility, navigate to the Data Rules tab and then the Oracle Dimension Sources subtab. Click the Undo button for the dimension rule.

Drill down into the Accounting Calendar to setup and execute the Calendar Rule process. This submission creates calendar hierarchy structure members (periods, quarters, etc.) in Enterprise Performance Foundation.

You can keep track of the submission history with the Last Run by and Last Run Date fields.

If you submit a dimension rule is with the incorrect mapping option specified, you can click the Undo button to reverse the effects of the dimension rule execution. You can then update the mapping and resubmit the dimension rule.

Note: Prior to undoing a dimension rule, any executed rules for ledgers using the chart of accounts must be reversed.

Execute Hierarchy Rules

The Hierarchy Rule Processing Engine populates Enterprise Performance Foundation with Oracle General Ledger hierarchical data as specified in the selected Hierarchy Rule. Though these hierarchies are not used during the balances transfer process, they are vital to subsequent analytical processing such as consolidation and budgeting.

Topic	Navigation Path
Execute Hierarchy Rules	Using the Enterprise Performance Foundation Administrator (or equivalent) responsibility, navigate to the Data Rules tab and then the Oracle Dimension Sources subtab. Drill down into the Hierarchy Rule for the relevant dimension to execute a particular version.

Hierarchy Rules

Chart of Accounts: **ADB Accounting Flex** Dimension: **Company Cost Center**
 Version: **Default Version** Organization: **Default Version**
 Effective Start Date: **01-Jan-1900** Version Description: **Default Version**
 Effective End Date: **01-Jan-2500**

Hierarchy Rules define how dimension hierarchies should be maintained.

Focus	Effective Start Date	Effective End Date	Add Version	Run	Update	Delete
▼ Default Version						
▼ MWH						
MWH	01-Jan-1900	01-Jan-2500				
▼ testrule						
testversion	01-Jan-1900	01-Jan-2500				

The process will first create any new parent dimension members that are part of the hierarchy, then proceed to creating the hierarchical parent-child relationships starting with the top parent value specified in the Hierarchy Rule.

Dimension values should already be defined in Enterprise Performance Foundation before any Hierarchy Rules are submitted.

Propagating changes in Hierarchy Structures

Parent-child relationships in a hierarchy are impacted by the following changes:

- Changes in the underlying segment hierarchy structure in Oracle General Ledger.
- New detail values (created in the associated chart of accounts segment value set) that fall into the existing hierarchy structure.

Hence, when you re-execute a Hierarchy Rule version, the process will always replace the existing parent child relationships in Enterprise Performance Foundation with the latest definition. If the existing hierarchy definition is already used by any Corporate Performance Management processes, then the process will not be able to replace the definition and will result in an error. In this case you will need to create a new version of the Hierarchy Rule.

Deleting Hierarchies

To delete hierarchies in Enterprise Performance Foundation that have been transferred from Oracle General Ledger through a Hierarchy Rule, use the following procedure:

1. Use the Executed Rule Request Removal program to undo the Hierarchy Rule. For information about using the Executed Rule Request Removal program, see Executed Rule Request Removal, page 8-7.
2. In Oracle General Ledger, navigate to the Configuration Rules tab, and then to the Dimension subtab. On the Hierarchy page, search for the appropriate hierarchy and click the **Update** icon on the parent row, then select **Read & Write** and click **Apply**.

Note: Only the user who ran the Hierarchy Rule can change the hierarchy to **Read & Write**.

3. Refresh the Hierarchy page in Enterprise Performance Foundation by searching for the hierarchy again. The **Delete** icon should now be enabled. Click the **Delete** icon for the hierarchy, then choose **Yes** in response to the confirmation prompt.
4. Remove the Hierarchy Rule that created the hierarchy that you have deleted.

Note: A Hierarchy Rule version can be deleted as long as it has not been executed to generate a hierarchy. A Hierarchy Rule in turn can only be deleted if it does not have an associated hierarchy rule version defined.

Execute Balances Rules

The Balances Rule Processing Engine transfers Oracle General Ledger balances data to Enterprise Performance Foundation. Dimension values, including accounting periods, should already be defined in Enterprise Performance Foundation before any Balances Rules are submitted.

Topic	Navigation Path
Execute Balances Rules	Using the Enterprise Performance Foundation Administrator (or equivalent) responsibility, navigate to the Data Rules tab and then the Oracle Data Sources subtab [Balances Rules Window] to run the appropriate Balances Rule version.

The screenshot shows the Oracle Enterprise Performance Foundation interface for configuring Balances Rules. The navigation menu includes Home, Data Rules (selected), Configuration Rules, Process Management, and Administration. Below the menu, there are links for Oracle Ledger Assignments, Oracle Dimension Sources, and Oracle Data Sources. The main content area is titled 'Balances Rules' and includes a tip: 'All Balances Rules will be stored in the folder Default.' A search form is present with fields for Ledger (Vision ADB (USA)), Rule (adb), and Effective Date (with an example of 25-Aug-2005). Below the search form is a 'Create' button and a table listing the created rule.

Focus Name	Start Date	End Date	Duplicate	Run	Update	Delete
Vision ADB (USA)						
adb						
adb	01-Jan-1900	01-Jan-2500				

Exception Processing

When transferring balances, if dimension values to which balances need to be posted do not exist in Enterprise Performance Foundation, then the Balances Rules process log file maintains a list of the Oracle General Ledger account code combinations that were not included in the balance transfer.

Miscellaneous Notes and Best Practices for Enterprise Performance Foundation and Oracle General Ledger Rules

Notes and best practices are discussed under the following headings:

- Balances Rules Considerations
- Dimensions Rules Considerations
- Hierarchy Rules Considerations
- Other Considerations

Balances Rules Considerations

Use a Single Balances Rule per Balance Type

Though Enterprise Performance Foundation allows for creation of multiple Balances Rules for each balance type (Actuals, Budget, or Encumbrance), Oracle recommends for Actual and Encumbrance balance types that you only define one Balances Rule per balance type. The use of a single rule will help ensure integrity in the base data that is used for reporting, modeling, and analysis in Enterprise Performance Foundation.

Note that you can create multiple versions of a Balances Rule to cater for changes in the rule definition.

Running Balances Transfer in Gapless and Sequential Period Order

It is your responsibility to ensure that there are no gaps in the range of periods for which balances are transferred to Enterprise Performance Foundation for a particular

ledger. The process will produce an error if it finds a missing period for which balances are not transferred.

If there has been backposting in Oracle General Ledger, the balances transfer needs to be re-run, beginning with the backposted period, and going forward to all subsequent periods. This is because Enterprise Performance Foundation does not incorporate any kind of roll forward logic but instead relies solely on the Oracle General Ledger balances information transferred for each period.

Oracle General Ledger Summary Account Balances

Balances stored in Oracle General Ledger summary accounts are not transferred to Enterprise Performance Foundation.

Distinguishing Standard, Average, and Statistical Balances

When transferring Oracle General Ledger balances into Enterprise Performance Foundation, the process leverages the Financial Element dimension to distinguish standard, average, and statistical balances. Specifically, the process will populate the Financial Element dimension based on the following conditions:

- For standard monetary balances, Financial Elements "Revenue" (display code 455), "Expense" (display code 457), and "End Balance" (display code 100) are used for Natural Accounts types Revenue, Expense, and Balance Sheet (for example, Liability, Assets, Owner's Equity) respectively.
- For average balances, Financial Element "Average Bal" (display code 140) is always used.
- For standard statistical balances, Financial Element "Statistic" (display code 10000) is always used. Oracle General Ledger does not track average statistical balances.

Output Data Set Requirements

Balances data for actual, budget, and encumbrance ledger balances for a given period can only be associated with one output data set. This ensures that these balances are backed by a single source of truth, rather than having the possibility of having multiple representations of the same data.

Note: For more information on the use of Datasets in General Ledger Rules, refer to Balances Output Dataset, page 12-22 earlier in the chapter.

Balances transfer for both budgets and encumbrance types will automatically result in unique data sets for these data types in each ledger.

For actual balances data however, it is possible to share a single output dataset across multiple ledgers. To track your actual balances in a consistent manner, we recommend that you create and maintain a separate output data set for actual balances for each ledger.

Dimension Rules Considerations

Dimension (and Hierarchy) Processing Frequency

To successfully transfer balances for all account code combinations, it is necessary for all account values and code combinations to be first mapped to Enterprise Performance Foundation.

General Ledger Rules architecture cannot automatically recognize changes and updates to Oracle General Ledger segment values and hierarchies. Hence, when values and hierarchies in the chart of accounts segments change, these changes are not automatically propagated in Enterprise Performance Foundation.

Note: This includes changes in Oracle General Ledger segment value descriptions and any attributes such as Natural Account Type assignments.

Depending on how frequently account values and hierarchies change in the chart of accounts in Oracle General Ledger, Dimension and Hierarchy Rules may need to be rerun before Balances Rules are submitted. If new child values are created in Oracle General Ledger, the execution of Dimension Rules will bring over new child values and Hierarchy Rules will bring over the updated parent values and hierarchy structure.

Ultimately, all relevant dimension members must be in Enterprise Performance Foundation to successfully transfer the Oracle General Ledger balances associated with a given account code combination. As such, the best business practice for managing the Oracle General Ledger data transfer to Enterprise Performance Foundation is to always submit the Dimension (and Hierarchy) Rules before running the Balances Rules.

Mapping of Dimensions to the Balancing Segment

There are three possible choices when it comes to mapping the Enterprise Performance Foundation Company Cost Center Organization Dimension to the Oracle General Ledger chart of accounts segments. You can choose to associate the balancing segment only, the cost center segment only, or the concatenation of both the company and cost center segment.

Financial data in Oracle General Ledger will only balance by balancing segment values. If you choose to associate the Company Cost Center Organization dimension with just the cost center segment, we recommend that you map another Enterprise Performance Foundation dimension to the balancing segment should the need arise to report on Enterprise Performance Foundation data that is in balance.

Account Type Assignments

The Extended Account Type attribute of the Natural Account and Line Item dimensions are set according to the Account Type specified in the Natural Account segment value which can be different from the Account Type assigned at the Oracle General Ledger account code combination level.

In Enterprise Performance Foundation, the dimension code combination's account type is strictly derived from the type associated with the natural account dimension member. This behavior is different from that in Oracle General Ledger and should be taken into account by users who utilize the architecture.

Enterprise Performance Foundation Value Sets

A value set is created in Enterprise Performance Foundation when one or more value sets from Oracle General Ledger chart of accounts segments are associated with an Enterprise Performance Foundation dimension via a Dimension Rule.

For example, consider two charts of accounts, each having a segment that uses the same value set in Oracle General Ledger. If both of these charts of accounts map this segment to the same dimension, then only 1 value set will be created for this dimension

in Enterprise Performance Foundation. However, if the charts of accounts map the segment to different dimensions, then separate value sets will be created for each of the dimensions in Enterprise Performance Foundation.

Hierarchy Rules Considerations

Child or Parent Account Values with Multiple Parents

Within an Oracle General Ledger segment values definition, it is possible to assign a child or parent segment value to multiple parents. When executing a Hierarchy Rule to transfer Oracle General Ledger segment value hierarchies into Enterprise Performance Foundation, the process will report an error if a child is assigned to multiple parents under the top parent specified in the rule. If you have such a case in your Oracle General Ledger hierarchies and you want to transfer them into Enterprise Performance Foundation, you need to define multiple Hierarchy Rules with the appropriate top parents such that the child will only be assigned to one parent for each rule.

Other Considerations

Integrity of Oracle General Ledger Set Up Data in Enterprise Performance Foundation

To maintain the integrity of Oracle General Ledger set up elements in Enterprise Performance Foundation such as calendar, value set members and value set hierarchies, it is important to only use the various General Ledger Rule constructs including Calendar Rules, Dimension Rules and Hierarchy Rules to propagate such data in Enterprise Performance Foundation. This will ensure a single source of truth for the set up data.

The architecture will not bar a user from using the Dimension Hierarchy Manager to make updates directly to these setup elements within Enterprise Performance Foundation. However, these changes will not propagate back to Oracle General Ledger. Consequently, by making such direct updates, Enterprise Performance Foundation's set up data will become inconsistent with that in Oracle General Ledger.

Functional Dimension Definition

This chapter covers the following topics:

- Introduction
- Creating Functional Dimension Definition Rules
- Searching for Functional Dimension Definition Rules
- Updating Functional Dimension Definition Rules
- Deleting Functional Dimension Definition

Introduction

This feature allows Corporate Performance Management users to employ a single Base dimension as multiple Data dimensions by applying different functional names to the Base dimension. Users can now specify multiple Functional Dimension Names for a single Base dimension and specify different data columns for each Functional Dimension Name.

For more information, please refer to the following topics

1. Creating Functional Dimension Definition Rules, page 13-2
2. Searching for Functional Dimension Definition Rules, page 13-3
3. Updating Functional Dimension Definition Rules, page 13-3
4. Deleting Functional Dimension Definition, page 13-4

The Functional Dimension Definition Process

The process for Functional Dimension Definitions is as follows:

1. Create a Functional Dimension Definition
2. View, modify, or delete a Functional Dimension Definition
3. Attach the created Functional Dimension Definition to the appropriate business process
4. Deleting Functional Dimension Definition

Accessing the Functional Dimension Definition

To access the Functional Dimension Definition feature:

1. Select the Enterprise Performance Foundation Administrator Responsibility

2. Select the Configuration option
3. Select Functional Dimension Definition on the submenu

Creating Functional Dimension Definition Rules

You can create Functional Dimension Definition Rule through Create Functional Dimension Definition wizard. Click Create button on the Functional Dimension Definition Summary Page to open the Create Functional Dimension Definition wizard

The Create Functional Dimension Definition Wizard consists of the following steps:

1. Create Rule
2. Create Functional Dimension Definition Names and Mapping
3. Review Definition and Mapping

Step 1: Create Rule

The first step in the Create Functional Dimension Definition wizard displays the “Definition” page.

1. In the folder box, select the folder in which you want to create the Functional Dimension Definition
2. In the **Rule Name** field, specify a name for the Functional Dimension Definition Rule
3. In the **Description** field, specify a test description for the Functional Dimension Definition rule
4. In the Access right option, select either **Read/Write** or **Read**.
5. Click **Next**

Step 2: Create Functional Dimension Definition Names and Mapping

The second step in the Functional Dimension Definition wizard displays the “Functional Dimension Definition Names and Mapping” page.

1. Using the button next to the **Dimension** field, select a dimension for which you want to define the mapping
2. In the **Functional Dimension Name** field, enter the name that you want to be displayed for the dimension that you have selected. By default, the default Dimension name appears in this field.
3. Click the **Update Table Column Display Name** check box, to also change the default column display name to the new Functional Dimension Name.
4. Click **Add Another Row** to add a new row to the Table and Column Mapping region
5. Using the button next to the **Table Name** field, select a table for which you want to define the column name.
6. Using the button next to the **Column Name** field, select a column name.
7. Click **Next**.

Step 3: Review Definition and Mapping

The last step in the Create Functional Dimension Definition Wizard displays the “Review Functional Dimension Names” page.

1. Review the Functional Dimension Definition rule mapping. If you want to make any modification to the definition, use the **Back** button to return to the step you want to change.
2. When you are satisfied that the Functional Dimension Definition mapping is correct, click **Finish** to create Functional Dimension Definition rule.

Searching for Functional Dimension Definition Rules

You can search for Functional Dimension Definition Rules from the Functional Dimension Definition Summary page. To perform a search, do the following,

1. (Mandatory) In the **Folder** box, use the drop-down list to select the folder in which you want to search.
2. In the **Name** field, type the name or part of the name of the Functional Dimension Definition Rule.

Note: If you leave this field blank, all of the rule in the folder will be displayed.

3. Click **Go** to display a list of Functional Dimension Rules that meet the search criteria.

Updating Functional Dimension Definition Rules

Use the following procedures to update a Functional Dimension Definition Rule:

1. On the Functional Dimension Definition Summary page, click the Update icon for the appropriate Functional Dimension Definition rule.
2. On the update page, make any desired changes to the Functional Dimension Definition Rule’s name, description, or access rights.
3. Click **Apply** to save your changes or **Cancel** to return to Functional Dimension Definition Summary page without saving your changes.

Updating Functional Dimension Definition Names and Mapping

Use the following procedure to update a Functional Dimension Definition Mapping

1. On the Functional Dimension Definition Summary page, click the Update icon for the appropriate Functional Dimension Definition Version. The
2. “Update Functional Dimension Details” page is displayed
3. In the **Functional Dimension Name** field, enter the full or partial name for which you want to search
4. Using the button next to the **Dimension** field, select a dimension for the Functional Dimension Name.
5. Click **Go** to view the Functional Dimension Names.

6. Click the “Details” icon next to the Functional Dimension Name you want to update. The “Update Functional Dimension Names and Mapping” page is displayed.
7. On the “Update Functional Dimension Names and Mapping” page, you can do the following
 - Make modification to the Functional Dimension Name field
 - Change the selections in the **Column Name** and **Table Name** fields

Deleting Functional Dimension Definition

On the Functional Dimension Definition Summary page, click the “Delete” icon for the appropriate Functional Dimension Definition. The Functional Dimension Definition rule and its definition and mapping are deleted.

Interface Tables and External Data Loaders

This appendix covers the following topics:

- Introduction
- Dimension Member Loader
- Dimension Hierarchy Loader
- External General Ledger Data Loader
- Detail Client Data Loader
- Dimension Loader
- Data Loader

Introduction

Enterprise Performance Foundation is an integral part of the Oracle E-Business Suite Unified Data Model. Enterprise Performance Foundation uses the product prefix FEM and is the central data depository upon which all applications are being built. This enables all applications to use a common data store.

This topic presents general introductory information about loading data into Enterprise Performance foundation. This appendix also contains detailed descriptions of the data loader programs.

Note: The DataX Loader program is being phased out. The Detail Client Data Loader program provides all of the functionality of the DataX Loader program as well as additional capabilities. Use the Detail Client Data Loader program instead of the DataX Loader for loading data into data tables.

For further information about the data loader programs, see the following:

- Dimension Member Loader, page A-7
- Dimension Hierarchy Loader, page A-52
- External General Ledger Data Loader, page A-72
- Detail Client Data Loader, page A-88
- Dimension Loader, page A-103
- Data Loader, page A-107

Loading Data into Enterprise Performance Foundation

Enterprise Performance Foundation provides an easy and automated way for loading and modifying your *business data* into the Enterprise Performance Foundation system. Business data, in this sense, refers to any information that describes your business operations, such as your customers, products, and revenues. The Enterprise Performance Foundation Interface Architecture is the gateway that enables you to get all of this information into your system, so that you can begin using the various Enterprise Performance Foundation tools and applications to better understand your business.

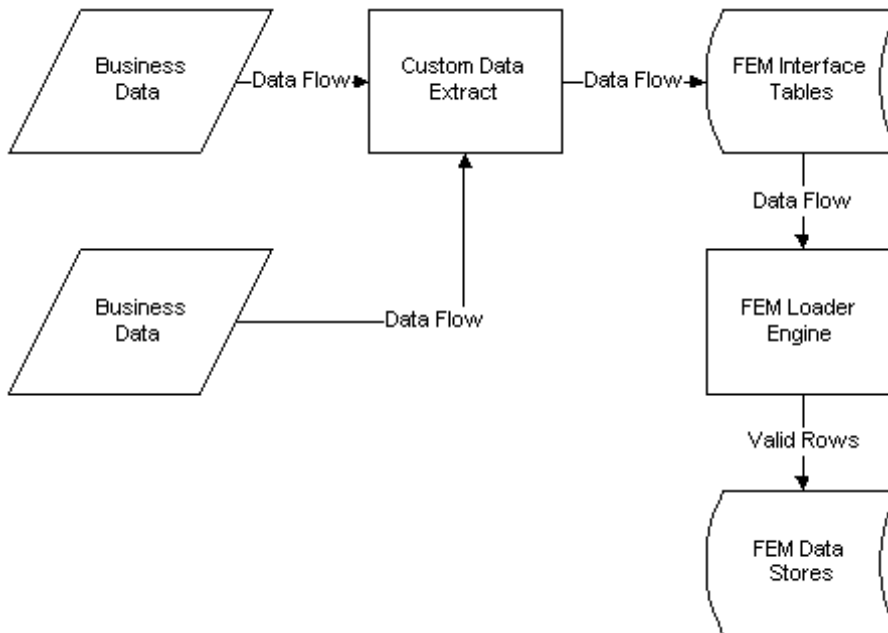
The Enterprise Performance Foundation Interface Architecture enables you to load the following types of information:

- Dimension Members and Attributes
- Dimension Levels
- Dimension Hierarchies
- Fact data, such as monetary balances and statistics
- General Ledger data
- Detail Client data

The Enterprise Performance Foundation Interface Architecture provides a set of *interface tables* for each type of information. An interface table acts as a staging area for your business data. The Enterprise Performance Foundation Interface Architecture also provides a set of *programs* that you can then use to move data from this staging area into the official Enterprise Performance Foundation database.

Interface Data Flow

The following diagram illustrates how business data flows from an external system into the Enterprise Performance Foundation database:



Interface Data Flow Definitions

The following table lists interface data flow definitions:

- **Business Data**
This is the data that describes business operations, such as customers, products, and revenues.
- **Custom Data Extract**
This is a process that moves the business data from the legacy systems (such as data entry systems) into the Enterprise Performance Foundation Interface tables.
- **Enterprise Performance Foundation Interface Tables**
These tables are a staging area for the business data before being moved into the official Enterprise Performance Foundation database for use by applications.
- **Enterprise Performance Foundation Loader Program**
The Loader Programs move data from the Interface tables into the official Enterprise Performance Foundation database.
- **Enterprise Performance Foundation Data Stores**
This is the *official* Enterprise Performance Foundation database. Data in the Enterprise Performance Foundation Data Stores is available for use by users and applications.

General Concepts

The following table lists general concepts of Enterprise Performance Foundation interface architecture and their definitions:

Concept	Definition
Dimension	A dimension is a structure that can be used to categorize business data. A dimension contains members. Enterprise Performance Foundation defines some dimensions (but not all) to be <i>hierarchical</i> , in that users can organize the members into one or more hierarchies. Examples of dimensions include Customer, Product, and Time.
Dimension Attributes	A dimension attribute is a property or qualifier that further describes a dimension member. An attribute may be anything, such as a Date, a number, or a character string. For example, the <i>Geography</i> dimension may have an attribute <i>Population</i> that designates how many people live in that area. Each member of the Geography dimension therefore has an associated population. Enterprise Performance Foundation considers Dimension Attributes to be a completely separate concept from Dimension Identifiers.

Concept	Definition
Dimension Attribute Versions	A dimension attribute version (also known simply as a <i>version</i>) is an identifier that distinguishes differing assignments for the same dimension attribute for a member. For example, as the population of the Geography member <i>Texas</i> changes over time, users track these changes by creating new versions for the <i>Population</i> attribute. <i>Version A</i> may represent the population in 1999, while <i>Version B</i> may represent the population in 2000.
Dimension Identifiers	A dimension identifier is a character string, or combination of character strings, that uniquely identifies each member of a dimension. Dimension identifiers are non-translatable, as they are the same regardless as to the language context. Each dimension has its own unique set of columns in the Enterprise Performance Foundation Interface tables that serve as the dimension identifier for that dimension.
Display Code	The display code is a non-translatable, alphanumeric string that is a component of the dimension identifier for most (but not all) dimensions.
Hierarchy Definition	A hierarchy definition is a structure of dimension members organized by parent/child relationships, for a designated effective date range. <i>Hierarchy definition</i> is synonymous with <i>hierarchy version</i> , in that it is one instance of the hierarchy.
Hierarchy Object	A hierarchy object is a collection of hierarchy definitions. The individual hierarchy definitions represent a particular picture of the hierarchy object.
Level	A level is a property of hierarchical dimensions that designates a category of like members. For example, in the Geography dimension there might be a level named <i>City</i> and a level named <i>State</i> . Geography members such as <i>Tulsa</i> and <i>Dallas</i> belong in the <i>City</i> level, while Geography members such as <i>Texas</i> and <i>Oklahoma</i> belongs in the <i>State</i> level. The designation of level is the same across all hierarchies within a dimension. In other words, <i>Texas</i> is always a state, in all Geography hierarchies.
Simple Dimension	A simple dimension is a dimension that does not have hierarchies or attributes. Simple dimensions are just a list of members.

Concept	Definition
Translatable Name	A translatable name is a user-friendly alphanumeric name that can vary by language. For example, a given dimension member may be referred to differently in French as compared to English. Each row that stores a translatable name (or description) has a language context that designates the language in which that name or description is viewed.
User-Configurable Dimension	A user-configurable dimension is a dimension that enables additional customization beyond the standard dimensions provided by Enterprise Performance Foundation. Enterprise Performance Foundation provides user-configurable dimensions that support hierarchies and attributes, as well as user-configurable <i>simple</i> dimensions.
Value Set	A value set is a set of dimension members. Value sets are a component of the dimension identifier for some dimensions. The value set concept enables larger organizations to employ the same code to identify multiple dimensions across different regions. For example, in Europe, code 123 may identify <i>Customer A</i> , while the same code in the United States identifies <i>Customer B</i> .

The Loading Process

The Enterprise Performance Foundation Interface Architecture provides a separate program to load different types of business information. Used together, these programs provide a way to load your business data into the Enterprise Performance Foundation system.

A typical loading process involves loading all types of information: members, hierarchies, fact, and general ledger data. Other than requiring you to always load dimension members first, the Enterprise Performance Foundation system does not require any particular order for the loads.

Loading Example

Use the following example of a typical loading process:

1. Load/update dimension members using the Dimension Member Loader.
2. Load/update hierarchies using the Dimension Hierarchy Loader.
3. Load generic fact data using the Detail Client Data Loader.
4. Load general ledger data using the External GL Data Loader.

You need to run only the loaders for which you have data. So in a given month, you may have no new dimension members. If so, you can skip running the Dimension Member Loader. Similarly, you may not have any general ledger data. If so, you would not run the External GL Data Loader program.

Troubleshooting Example

For any errors encountered by the loaders, use the following example of a troubleshooting process:

1. Review the Concurrent Manager log for any messages produced by the programs.
2. Review the STATUS column for each row in the interface _T table. For descriptions of the STATUS column status codes, refer to the output information in the topic on running the loader in the appropriate section of this appendix.

The following sections provide information on how to prepare your data for loading as well as how to run each of these programs.

Loader Dimension Defaults

In some situations, you may want to use a default member to populate a dimension column in an interface table. For example, you may want to populate *Not Applicable* for a dimension column in one of the FEM_DATAx_T tables because you are not using that dimension in your data but it is included in the defined dimensionality (that is, the processing key) of the target FEM_DATAx table. Another example is when you are loading new dimension members but you do not know what to populate for some of the required attributes. In this case, you would like to use a default member for the attribute dimension.

Defaults for Attribute Dimensions

For dimensions employed as an attribute list of value (such as Source System or Flag), Oracle provides a seeded default member for the dimension. You can modify this default from the *Dimension: Definition* page.

When loading new dimension members, the Dimension Member Loader requires an assignment for each required attribute of that dimension. Oracle recommends that you attempt to find a meaningful assignment value for these attributes as much as possible. In some situations, however, you may want to default the attribute assignment to the default member for the attribute dimension.

To do this, you need to identify the *attribute* dimension for the required attribute. You can do this from Dimension and Hierarchy Management by looking up the required attributes for the dimension. Not all attributes have an *attribute* dimension. Rather, only those where the Attribute Data Type is set to 'DIMENSION' require a value from another dimension as the attribute assignment. For these, you can identify that dimension and then look up the default member for it from the *Dimension: Definition* page.

For *Flag* attributes (that is, those required a 'Y' or an 'N' assignment), Oracle recommends that you do not use the default member because that would mean that all of your flag attributes would have a meaningless assignment. For Flag attributes, Oracle recommends that you attempt to assign a meaningful value, as either 'Y' or 'N', because those are the only two possibilities for such attributes.

The Dimension Member Loader does not refer to these attribute defaults directly. Rather, you refer to the defaults when you populate the appropriate _ATTR_T interface table with attribute assignments for your new members.

Defaults for Value Set Dimensions

For Value Set dimensions, Enterprise Performance Foundation generates default members for each Value Set created in the system. Once these default members

are generated, you can reference them when you populate your FEM_DATAx_T interface tables. The Detail Client Data Loader does not directly reference these defaults. Rather, you use these default members to populate dimension columns in the FEM_DATAx_T tables for which you do not have a meaningful value.

Enterprise Performance Foundation does not require that you populate a value for each dimension column in the FEM_DATAx_T tables. Rather, it requires only that you populate dimension columns that are either not null or part of the processing key defined for the target FEM_DATAx table

Using Concurrent Manager

The External Loaders require only that you specify the parameters for the load. All other Concurrent Manager pages (for scheduling and others) are optional. For each of these other steps, you can press *Next* to bypass them. You need to specify information in these other pages only if you are implementing the designated Concurrent Manager functionality. Refer to the appropriate Concurrent Manager documentation in the Oracle Applications User's Guide and Oracle Applications System Administrator's Guide for information on these other pages.

Folder Requirement

To successfully run any of the loader programs, the user must have the Data Integration folder assigned.

Dimension Member Loader

Use the Dimension Member Loader to load levels, dimension members, and their attributes. This program supports the following types of loading operations:

- Creation of new levels
- Update of translatable names/descriptions for existing levels
- Update of level sequence and the time group type (for Calendar Period only) for existing levels
- Creation of new Dimension Members and attribute assignments for those new members
- Update of translatable names/descriptions for existing Dimension Members
- Update of existing attribute assignments for existing Dimension Members
- Loading of new versions of attribute assignments for existing Dimension Members

Tip: You can use the Dimension Member Loader to update the Enabled Flag attribute for members. This is the easiest way to update the Enabled Flag for a large number of members.

Dimensions Supported for Loading

The Dimension Member Loader supports the loading of a pre-defined set of dimensions. It is not possible to load any other dimensions using this loader. As the structure of each of these dimensions varies, they can be separated into the following categories:

Time Dimension (Attributed)

- Calendar Period

Value Set Dimensions (Attributed)

- Channel
- Company Cost Center Organization
- Customer
- Entity
- Geography
- Line Item
- Natural Account
- Product
- Project
- Task
- Ten User Defined Hierarchical Dimensions

Simple Dimensions

- 200 User-Configurable Generic Simple Dimensions

Note: The *Attributed* label means that a dimension requires attribute assignments for new members to be valid. Each dimension has its own set of *required* attributes for which the user must provide assignments for the new member to be loaded.

Time Dimension (Attributed)

- The Calendar Period dimension is the only dimension in this category.

Note: For the Calendar Period dimension, the Number of Processes setting on the Multiprocessing: Parameters page is limited to a value of 1, because setting a value greater than 1 could cause data integrity violations for Calendar Period dimension members. For any other dimension, you can increase the number appropriately for your system to take advantage of the multiprocessing capabilities of the loader.

Identifying Calendar Period Members

Calendar Period is the only dimension whose identifier is a combination of the four separate components:

- Calendar identifier
- Level identifier
- End Date
- Period Number

The combination of the four components guarantees uniqueness for each Calendar Period member within the interface tables.

When creating new Calendar Period dimension members, or updating the properties and attributes for existing Calendar Period members, specify time in the interface tables using the following columns:

- CALENDAR_DISPLAY_CODE (Calendar)

Each Calendar Period member belongs to one and only one Calendar. Typically, an installation has only a single Calendar defined for the entire system. In larger implementations, however, it is possible to have multiple Calendars defined. Therefore, you always need to provide a Calendar when identifying members of the Calendar Period dimension. Oracle provides a seeded Calendar named *Default Calendar* that you can use.

- DIMENSION_GROUP_DISPLAY_CODE (Level)

This is the Level to which the Calendar Period belongs. For example, Calendar Periods spanning 28, 30, and 31 days would be designated as belonging to the *Month* level. Calendar Periods spanning 24 hours would be designated as belonging to the *Day* level.

- CAL_PERIOD_END_DATE (End Date)

This is the end date of the calendar period. For example, to designate the *March* period for the year 2004, the End Date is '31-MAR-2004'.

- CAL_PERIOD_NUMBER (Period Number)

This is a relative number of the Calendar Period within a given Fiscal Year. The Period Number is a unique number that designates the sequential order of the Calendar Periods for periods of the same Level.

For example, if a company's Fiscal Year starts in June and ends in May the following year, and every period starts on the first of the month and ends on the last day of the month, there would be the following relationships among the Fiscal Year, Level (Dimension Group Display Code), End Date, and Period Number:

Module Fiscal Year	Period Name	Level (Dimension Group)	End Date	Period Number
2004	May-FY04	Month	31-May-2004	12
2005	Jun-FY05	Month	30-Jun-2004	1
2005	Jul-FY05	Month	31-Jul-2004	2
2005	Aug-FY05	Month	31-Aug-2004	3
2005	Sep-FY05	Month	30-Sep-2004	4
2005	Oct-FY05	Month	31-Oct-2004	5
2005	Nov-FY05	Month	30-Nov-2004	6
2005	Dec-FY05	Month	31-Dec-2004	7
2005	Jan-FY05	Month	31-Jan-2005	8
2005	Feb-FY05	Month	28-Feb-2005	9
2005	Mar-FY05	Month	31-Mar-2005	10
2005	Apr-FY05	Month	30-Apr-2005	11
2005	May-FY05	Month	31-May-2005	12
2005	Qtr1-FY05	Quarter	31-Aug-2004	1
2005	Qtr2-FY05	Quarter	30-Nov-2004	2
2005	Qtr3-FY05	Quarter	28-Feb-2005	3
2005	Qtr4-FY05	Quarter	31-May-2005	4
2005	Year-FY05	Year	31-May-2005	1
2006	Jun-FY06	Month	30-Jun-2005	1

Calendar Period Attributes

You must provide assignments for all of the Calendar Period required attributes for each new member to be loaded. The list of required attributes for the Calendar Period dimension, along with the required attributes for all dimensions, is provided in the *Dimension Attributes* section of this document.

Value Set Dimensions (Attributed)

A Value Set dimension is a dimension whose identifier includes Value Set as a component.

Identifying Value Set Dimension Members

You identify members for these dimensions using both an alphanumeric code for the member (also known as the *Member Display Code*) in combination with an alphanumeric

code for the Value Set (also known as the *Value Set Display Code*). For example, the identifiers for the Product dimension are the following columns:

- PRODUCT_DISPLAY_CODE
- VALUE_SET_DISPLAY_CODE

Typically, there is only a single Value Set for all Product members. It is possible, however, for large implementations to employ multiple value sets for a dimension. Therefore, you must always provide the VALUE_SET_DISPLAY_CODE to uniquely identify a member of these dimensions.

You must provide assignments for any required attributes of that dimension for each new member to be loaded. The set of required attributes varies from dimension to dimension. Refer to the *Dimension Attributes* section for information on the required attributes for each dimension.

Identifying Value Sets for the Dimension Member Loader interface tables

The Dimension Member Loader interface tables require that you be able to properly identify the value set when loading a member in a value set dimension. A value set dimension is defined as a dimension that employs Value Set to differentiate dimension members. The value set dimensions are listed as follows:

- Channel
- Company
- Company Cost Center Organization
- Cost Center
- Customer
- Entity
- Financial Element
- Geography
- Line Item
- Natural Account
- Product
- Project
- Task
- User Dimension1
- User Dimension2
- User Dimension3
- User Dimension4
- User Dimension5
- User Dimension6
- User Dimension7
- User Dimension8

- User Dimension9
- User Dimension10

For example, if you are loading members for the Product dimension, you need to identify the target value set for each member. Additionally, you need to identify the value set of any attribute assignment when the assignment is a member of a value set dimension. This is true for each dimension in the value set dimensions list.

You identify value sets in the Dimension Member interface tables using the non-translatable alphanumeric identifier, also known as the *Display Code*. Every value set in the system has such an identifier. Once you know the display code of the value set that you want to use, you populate either the VALUE_SET_DISPLAY_CODE or ATTR_ASSIGN_VS_DISPLAY_CODE column in the appropriate interface table, depending upon what type of operation you are performing.

For example, when you are loading new product members, you identify the appropriate Product value set in the VALUE_SET_DISPLAY_CODE column in the FEM_PRODUCTS_B_T, FEM_PRODUCTS_TL_T, and FEM_PRODUCTS_ATTR_T tables. This is true for loading new members, updating existing members, or loading/updating attribute assignments for those members.

When you are loading in an attribute assignment that contains a member from a value set dimension, you populate the ATTR_ASSIGN_VS_DISPLAY_CODE column in the appropriate ATTR_T table. For example, when an attribute requires that the assignment originate from a value set dimension (such as PRODUCT), you need to populate the ATTR_ASSIGN_VS_DISPLAY_CODE column for that attribute assignment so that the product member is properly identified.

Oracle provides an initial *starter* value set to use for each value set dimension in the system. Implementations that do not employ multiple value sets for any dimension normally use these starter value sets. The following table lists each of the value set dimensions in the system along with the corresponding display code of the Oracle starter value set:

Dimension Name	Value Set Display Code
Channel	CHANNEL
Company	COMPANY
Company Cost Center Organization	COMPANY_COST_CENTER_ORG
Cost Center	COST_CENTER
Customer	CUSTOMER
Entity	ENTITY
Financial Element	FINANCIAL_ELEMENTS
Geography	GEOGRAPHY
Line Item	LINE_ITEM
Natural Account	NATURAL_ACCOUNT
Product	PRODUCT
Project	PROJECT
Task	TASK
User Dimension1	USER_DIM1
User Dimension2	USER_DIM2
User Dimension3	USER_DIM3
User Dimension4	USER_DIM4
User Dimension5	USER_DIM5
User Dimension6	USER_DIM6
User Dimension7	USER_DIM7
User Dimension8	USER_DIM8
User Dimension9	USER_DIM9
User Dimension10	USER_DIM10

If your implementation uses only the starter value sets, then you can rely on the starter set values sets for each Value Set dimension list to identify the display code for each value set in the system. If your implementation employs value sets other than those that Oracle provides at installation, however, you need to identify the appropriate value set or sets for your dimension using the Administration application.

The following lists both summarize and provide detailed instructions of the process of identifying the appropriate value set for each dimension.

To summarize the process, do the following:

1. Identify a Ledger that is valid for the dimension member.
2. Identify the Global Value Set Combination for that Ledger.

3. Identify the Value Sets for each dimension in the Global Value Set Combination.
4. Find the display code for each Value Set.

To perform detailed instructions, do the following:

1. Identify the Ledger in which you are working from your Application Preferences. Each Ledger has one and only one value set assigned for each of the dimensions in the value set dimensions list. The ledger chosen in the Application Preferences for each user has only one associated global value set combination.
2. Go to the *Dimension: Members* page and switch the dimension perspective to the Ledger dimension.
3. Once you have identified the appropriate Ledger, click on it to view the attribute assignments.
4. Record the value for the Ledger Global Value Set Combination attribute. You use this value in the next step to identify the individual value sets of your Ledger.
5. Navigate to the *Registration: Global Value Set Combinations* page and select the Global Value Set Combination identified in the previous step. You can select it by clicking on the appropriate *Update* link. This page displays the various Value Sets for each dimension. However, the page identifies the Value Sets by an internal number concatenated with the translatable name instead of the display code. For example, a value set might be identified as *13 – Channel*. The *13* is an internal number identifier for the value set. The *Channel* is the translatable name for that value set.
6. To find out the display code for each Value Set, navigate to the *Registration: Value Sets* page and search on the appropriate Value Set name from the previous step. For example, if the value set was identified as *13 – Channel*, then you search on the Value Set name of *Channel*.
7. Note that the value in the *Code* column of the search results is the display code of the value set. This is the value for populating the `VALUE_SET_DISPLAY_CODE` column in the appropriate Dimension Member Loader interface tables.

Because it is possible for multiple Ledgers to share the same Global Value Set Combination, the value sets assigned to each dimension are identical for those Ledgers. In this situation, it does not matter which of the ledgers you choose to identify the Global Value Set Combination.

Simple Dimensions

Simple Dimensions do not have attributes. The only information that you need to provide for simple dimensions are an identifier and a translatable name.

Identifying Simple Dimension Members

You identify simple dimension members with a single *Member Code* value. This value is a non-translatable alphanumeric identifier for the member. Because the program loads members for all simple dimensions from a single interface table, however, you must also provide an identifier for the Dimension with each.

For example, populate the following columns to identify a new simple dimension member:

- `MEMBER_CODE`
- `DIMENSION_VARCHAR_LABEL`

Populating the Interface Tables for the Dimension Member Loader

The following section provides information on how to populate the Dimension Member Loader interface tables in preparation for a load.

Levels

Populate the following interface tables for loading Levels:

- FEM_DIMENSION_GRPS_B_T
- FEM_DIMENSION_GRPS_TL_T

The FEM_DIMENSION_GRPS_B_T table is the identifying table for Levels. Populate this table with a single record for each distinct Level. The DIMENSION_GROUP_DISPLAY_CODE is the non-translatable alphanumeric code that identifies each Level. The DIMENSION_VARCHAR_LABEL is the identifier for the dimension to which the Level belongs.

When defining a Level in the FEM_DIMENSION_GRPS_B_T and FEM_DIMENSION_GRPS_TL_T tables, you need to provide a value for the DIMENSION_GROUP_SEQ of each Level. This is a sequence number that defines the hierarchical relationship of the Levels, with the smallest number representing the *top* of the hierarchy and the largest number representing the *bottom*. For example, assuming two Levels, *Country* and *City*, with *Country* having a DIMENSION_GROUP_SEQ = 1 and *City* having a value of 2, the members in the *Country* Level are parents of members in the *City* Level.

The FEM_DIMENSION_GRPS_TL_T table is the translatable table for Levels. Populate this table with a record for each installed language for each distinct Level. If you have just one language installed in your system, then you populate this table with a single record per Level. If you have ten languages installed in your system, populate this with a record for each installed language.

The program does not require language rows for any of the additional installed languages. If you do not provide such rows, the loader defaults the name and description for the other languages to be the same as the name and description for the record of the current session. For example, you have ten installed languages, but have translations for your Levels in only six languages. This means that you would like to have language rows for the other four languages, but it is just that you have not actually translated the Level names into those languages yet. In this case, you provide the four language rows per Level in the FEM_DIMENSION_GRPS_TL_T table. When the loader runs, it creates the new Levels with the specified names in those four languages. The loader also creates rows for the other four languages by defaulting the name and description to the name and description of the current session language.

Dimension Members

You need to populate the appropriate set of interface tables before running the loader to move the members into Enterprise Performance Foundation. The interface tables that you populate depend upon the dimension that you are loading and the type of load that you are performing.

The following matrix lists the various member interface tables for each dimension:

Dimension	Member Table	Translatable Table	Attribute Table
Calendar Period	FEM_CAL_PERIODS_B_T	FEM_CAL_PERIODS_TL_T	FEM_CAL_PERIODS_ATTR_T
Channel	FEM_CHANNELS_B_T	FEM_CHANNELS_TL_T	FEM_CHANNELS_ATTR_T
Company Cost Center Organization	FEM_CCTR_ORGS_B_T	FEM_CCTR_ORGS_TL_T	FEM_CCTR_ORGS_ATTR_T
Customer	FEM_CUSTOMERS_B_T	FEM_CUSTOMERS_TL_T	FEM_CUSTOMERS_ATTR_T
Entity	FEM_ENTITIES_B_T	FEM_ENTITIES_TL_T	FEM_ENTITIES_ATTR_T
Geography	FEM_GEOGRAPHY_B_T	FEM_GEOGRAPHY_TL_T	FEM_GEOGRAPHY_ATTR_T
Line Item	FEM_LN_ITEMS_B_T	FEM_LN_ITEMS_TL_T	FEM_LN_ITEMS_ATTR_T
Natural Account	FEM_NAT_ACCTS_B_T	FEM_NAT_ACCTS_TL_T	FEM_NAT_ACCTS_ATTR_T
Product	FEM_PRODUCTS_B_T	FEM_PRODUCTS_TL_T	FEM_PRODUCTS_ATTR_T
Project	FEM_PROJECTS_B_T	FEM_PROJECTS_TL_T	FEM_PROJECTS_ATTR_T
Task	FEM_TASKS_B_T	FEM_TASKS_TL_T	FEM_TASKS_ATTR_T
User Dim1	FEM_USER_DIM1_B_T	FEM_USER_DIM1_TL_T	FEM_USER_DIM1_ATTR_T
User Dim2	FEM_USER_DIM2_B_T	FEM_USER_DIM2_TL_T	FEM_USER_DIM2_ATTR_T
User Dim3	FEM_USER_DIM3_B_T	FEM_USER_DIM3_TL_T	FEM_USER_DIM3_ATTR_T
User Dim4	FEM_USER_DIM4_B_T	FEM_USER_DIM4_TL_T	FEM_USER_DIM4_ATTR_T
User Dim5	FEM_USER_DIM5_B_T	FEM_USER_DIM5_TL_T	FEM_USER_DIM5_ATTR_T
User Dim6	FEM_USER_DIM6_B_T	FEM_USER_DIM6_TL_T	FEM_USER_DIM6_ATTR_T
User Dim7	FEM_USER_DIM7_B_T	FEM_USER_DIM7_TL_T	FEM_USER_DIM7_ATTR_T
User Dim8	FEM_USER_DIM8_B_T	FEM_USER_DIM8_TL_T	FEM_USER_DIM8_ATTR_T
User Dim9	FEM_USER_DIM9_B_T	FEM_USER_DIM9_TL_T	FEM_USER_DIM9_ATTR_T

Dimension	Member Table	Translatable Table	Attribute Table
User Dim10	FEM_USER_DIM10_ B_T	FEM_USER_DIM10_ TL_T	FEM_USER_DIM10_ ATTR_T
Generic Simple Dimensions	FEM_SIMPLE_DIMS_ B_T	FEM_SIMPLE_DIMS_ TL_T	N/A

Member Table

A single row in the Member table identifies each dimension member. The Member table is also where you specify the Level for each member.

Translatable Table

Populate the Translatable table with a row for each dimension member per installed language in the system. If only one language is installed in your system, then you need only a single row per dimension member in the Translatable table. If multiple languages are installed, however, the loader enables you to specify a row per each installed language so that the name and description of each member can be translated for that language.

Note: In a multi-language system you are not required to provide a row for each installed language in the Translatable table. If you do not provide a row for each language, the loader defaults the name for any missing languages to the name of the member in the Translatable table for the language of your current session. You must, however, always provide a row in the Translatable table in the language of your current session for each new member.

Attribute Table

Populate the Attribute table with a row for each attribute assignment for the member. New member loads require an assignment row for each required attribute of that dimension. Update loads do not require any assignment rows.

The ATTRIBUTE_VARCHAR_LABEL column identifies the attribute of each assignment row. You can obtain a list of valid values for the ATTRIBUTE_VARCHAR_LABEL from the *Dimension: Attributes* page. You need the information in the Code Column.

The VERSION_DISPLAY_CODE identifies the version of the attribute assignment. For new members, the specified version must be the designated *default* version of the attribute. Each attribute can have only a single such version defined in the Attribute metadata. In addition, when the attribute is *non-versionable*, then there is only a single valid VERSION_DISPLAY_CODE for any assignments of that attribute.

For all non-versionable seeded attributes (which is most if not all of the seeded attributes), use the Oracle provided seeded VERSION_DISPLAY_CODE of *Default* to designate the version for the attribute assignment.

Attribute assignments can be one of four possible data types:

- Dimension Member

Use the Display Code of the attribute dimension when providing an assignment that is a member from a dimension.

- Number

This is for free-form number attribute assignments (example – *Population*)

- Varchar

This is for free-form alphanumeric attribute assignments

- Date

Provide the assignment in the same format as specified by the *Date Format Mask* input parameter. For example, if the input *Date Format Mask* is 'YYYY/MM/DD', then an example date assignment is '1998/12/31'.

Loader Operations for the Dimension Member Loader

This section describes the various types of loads that you can perform with the Dimension Member Loader.

Note: The Dimension Member Loader performs all of the described types of loads in every run. This means that you can both create new members and levels as well as update existing members and levels in the same run. The actual source data in the interface tables determines what operations the loader actually performs. For example, if you have provided records for new levels as well as new members, then the loader performs both the *Create New Levels* and the *Create New Members* operations.

Create New Levels

The loader enables you to create new levels. Populate the following tables for a new level load:

- FEM_DIMENSION_GRP_S_B_T
- FEM_DIMENSION_GRP_S_TL_T (for level translatable names/descriptions)

You need only a single record in the Translatable table (for the language of your session) to create a valid new level. If you populate the translatable table with rows for other installed languages, however, the loader also loads these into the system.

Update Level Names and Descriptions

The loader enables you to modify the translatable name and description for any existing level. Populate the following table for an *Update Level Name* load:

- FEM_DIMENSION_GRP_S_TL_T

You do not need to populate any other tables for such a load.

Create New Members

The loader enables you to create new dimension members for any of the supported dimensions. Populate the following tables for a *New Member* load:

- Member Table
- Translatable Table
- Attribute Table (if applicable)

When loading new Budget members, the loader requires an assignment for the BUDGET_FIRST_PERIOD and BUDGET_LAST_PERIOD attributes. Each of these

attributes requires a valid Calendar Period member as the assignment. At this time, the only way to designate the Calendar Period is by the 32-digit numeric identifier. You can obtain this identifier by querying from the *Member* tab in the Dimension and Hierarchy Manager for the Calendar Period that you want to use. You need to update the ATTRIBUTE_ASSIGN_VALUE column in the FEM_BUDGETS_ATTR_T table with the appropriate Calendar Period numeric identifier for both the BUDGET_FIRST_PERIOD and BUDGET_LAST_PERIOD attributes.

Attribute assignments are not applicable to all dimensions. You need only a single record in the Translatable table (for the language of your session) to create a valid new member. If you populate the Translatable tables with rows for other installed languages, however, the loader also loads these into the system.

Update Member Names and Descriptions

The loader enables you to modify the translatable name and description for any existing member. Populate the following table for an *Update Member Name* load:

- Translatable Table

You do not need to populate any other tables for such a load.

Update Assigned Level of Existing Members (non-Time dimensions only)

The loader enables you to modify the level that has been assigned to any existing members. Populate the following table for an *Update Assigned Level* load:

- Member Table

You do not need to populate any other tables for such a load. To update the level for a member, populate the table with a row for each existing Value Set/Member Display Code combination that you want to update. Then specify the value in the DIMENSION_GROUP_DISPLAY_CODE column to which you want the member assigned.

Update Existing Attribute Assignments

The loader enables you to modify existing attribute assignments. Populate the following table for an *Update Attribute Assignment* load:

- Attribute Table

You do not need to populate any other tables for such a load. The loader checks that the designated member has an existing assignment for the ATTRIBUTE_VARCHAR_LABEL and VERSION_DISPLAY_CODE combination. If it does, it updates the assignment with the new assignment value from the Attribute interface table.

Note: For attributes that allow multiple assignments, the loader always creates a new assignment for the same version, rather than updating any existing assignments. In these situations, it is not knowable whether the user wants to create a new assignment or update an existing assignment because there are multiple possible assignments from which to choose.

Create New Attribute Assignment Versions

The loader enables you to load new versions of attribute assignments for existing members. This type of load is available only for *versionable* attributes (that is, those attributes that allow multiple versions). In addition, this type of load is only for the

assignments. It requires you to pre-define new versions in the Enterprise Performance Foundation metadata before running the load. This can be done in the Dimension Attribute page in Administration. Populate the following table for a *Create New Attribute Assignment Versions* load:

- Attribute Table

You do not need to populate any other tables for such a load. The loader checks that the attribute (identified in `ATTRIBUTE_VARCHAR_LABEL`) allows multiple versions, however. If it does, the loader loads any assignments for the new versions as long as the versions are defined in the Enterprise Performance Foundation metadata.

Running the Dimension Member Loader

This section describes how to launch the loader from Concurrent Manager.

Concurrent Program

Choose the *Dimension Member Loader* concurrent program.

Restrictions

The Dimension Member Loader does not allow multiple users to load levels, members, or attributes from the same dimension at the same time. Users can simultaneously run the loader, however, as long as they are loading separate dimensions.

Executed Rules

You cannot remove any level, members, or attribute assignments loaded from the Dimension Member Loader. The loader does enable you to update any property, name, or value associated with the loaded levels, member, and attribute assignments. You can therefore correct most erroneous data by performing a subsequent update operation for the same level, member, or attribute.

If you need to permanently remove the level, member, or attribute assignment, you must use the Dimension and Hierarchy Management interface to perform such deletes.

Note: Enterprise Performance Foundation does not support the physical deletion of levels or members. Enterprise Performance Foundation logically deletes these identifiers to preserve the integrity of possible historical data in the system.

Input Parameters

The Dimension Member Loader requires the following input parameters:

- For Dimension, pick a dimension from the pick list. The loader loads only those levels and members of the designated dimension.
- For Execution Mode, pick an execution mode of either *Snapshot* or *Error Reprocessing* from the pick list. If you pick *Snapshot*, the loader loads all records where the `STATUS = 'LOAD'`. If you pick *Error Reprocessing*, the loader loads only those records where the `STATUS <> 'LOAD'`.

Note: Status values must all be in upper case. If not, the loader programs may not pick up all the rows.

Output

The loader removes any levels, members, and attribute assignments that it was able to successfully load.

In addition to the standard Concurrent Manager error and log messages, the loader also updates the STATUS column for error rows in the source interface tables with a code indicating what was wrong with the record.

Status Codes	Meanings
DIMENSION_GROUP_SEQ_NOT_UNIQUE	There are duplicate group sequences in the interface table. Dimension Group sequences must be unique within a given dimension.
DIMGRP_NAME_EXISTS	The Dimension Group Name already exists for a different Dimension Group.
GROUP_EXISTS	The Dimension Group already exists in the Enterprise Performance Foundation database so it was not imported.
INVALID_ATTRIBUTE_LABEL	The value for ATTRIBUTE_VARCHAR_LABEL that identifies the attribute in the _ATTR_T table is invalid.
INVALID_CAL_PERIOD_START_DATE	The Start Date for a Calendar Period is after the End Date. The Start Date must always be before the End Date.
INVALID_DATE	A DATE attribute assignment value is invalid.
INVALID_DIM_ASSIGNMENT	The attribute assignment value is invalid.
INVALID_DIMENSION_GROUP	The Dimension Group does not exist in the Enterprise Performance Foundation database.
INVALID_DIMENSION_LABEL	The Dimension Label does not exist in Enterprise Performance Foundation.
INVALID_GL_PERIOD_NUM	The value for the GL_PERIOD_NUM attribute is invalid.
INVALID_GROUP_SEQ_ENFORCED_CODE	The Group Sequence Enforced Code is invalid.
INVALID_LVL_SPEC_ATTR_LABEL	The attribute being loaded is a level-specific attribute that has been assigned to a level to which the current member does not belong.
INVALID_MEMBER	The dimension member does not exist in Enterprise Performance Foundation.
INVALID_NUMBER	A NUMBER attribute assignment value is invalid.
INVALID_OBJ_DEF_ID	The Object Definition ID input parameter does not exist in Enterprise Performance Foundation.
INVALID_REQUIRED_ATTRIBUTE	One of the required attributes for a dimension member is invalid, thus preventing import of the dimension member.
INVALID_VALUE_SET	The Value Set does not exist in Enterprise Performance Foundation.

Status Codes	Meanings
INVALID_VERSION	The VERSION_DISPLAY_CODE does not identify an existing Attribute Version.
LOAD	The record is ready for loading.
MEMBER_IS_READ_ONLY	The attribute cannot be loaded because the dimension member is designated as read-only.
MISSING_NAME	The record is a <i>new</i> member or group that does not have a corresponding TL_T record.
MISSING_REQUIRED_ATTRIBUTE	The dimension member to be created is missing a required attribute in the appropriate attribute interface table.
MULT_VERSION_NOT_ALLOWED	Assignments in multiple versions are not allowed for this attribute.
OVERLAP_EXIST_START_DATE	The start/end date range for a calendar period in the interface table conflicts with the start/end date range for an existing calendar period.
OVERLAP_START_DATE_IN_LOAD	The start/end date ranges for two calendar periods in the interface table conflict with one another.
PROTECTED_ATTR_ASSIGN	The attribute assignment value is protected and cannot be changed.

Dimension Attributes

This section describes the attributes for each of the *Attributed* dimensions.

Attributes initially set to *Required* at installation are designated by a 'Y' value in the *Req* column. It is possible, however, for the user to configure the attributes in Enterprise Performance Foundation so that they are required only for members at certain levels. Therefore, any attributes designated as *Required* could potentially be optional in your particular installation if they have been configured that way in Dimension and Hierarchy Management.

The *Label* is the value that you specify in the ATTRIBUTE_VARCHAR_LABEL column of the appropriate FEM_xName_ATTR_T interface for the attribute assignment row.

Because users may define their own attributes using Dimension and Hierarchy Management, the labels for such attributes are not included in the list. It is not possible for users to define *required* attributes, so any such user-defined attributes are always optional when creating a new member.

Calendar Period

Attribute	Required	Label	Attribute Description
Calendar Period Accounting Year	Y	CAL_PERIOD_ACCOUNTING_YEAR	Identifies the Accounting year that applies for the period. For example, if the Accounting year is May to May, then the date June-30 2002 has an Accounting year of 2003.
Calendar Period Start Date	Y	CAL_PERIOD_START_DATE	The first date of the period defined by the Time dimension member.
Calendar Period End Date	Y	CAL_PERIOD_END_DATE	The last date of the period defined by the Time dimension member.
Calendar Period Adjustment Period Flag	Y	CAL_PERIOD_ADJ_PERIOD_FLAG	Identifies if the dimension member is an Adjustment Period. A 'Y' indicates that the period is an adjustment period. An 'N' indicates it is not.
Calendar Period Current Period Flag	Y	CAL_PERIOD_CUR_PERIOD_FLAG	Identifies if the dimension member is the Current Period. A 'Y' indicates that the period is the "Current Period". Only one period within a Level should be designated as the <i>Current Period</i> .
Calendar Period GL Origin Flag	Y	CAL_PERIOD_GL_ORIGIN_FLAG	Designates if the Call Period originally came from GL. All periods imported from a GL Calendar are automatically assigned 'Y' for this attribute. In addition, periods cloned by the hierarchy editor that originally came from GL are also assigned 'Y' for this attribute. Only new periods created in the Hierarchy Editor that never originated from GL do not receive a 'Y' assignment for this attribute.

Attribute	Required	Label	Attribute Description
Calendar Period GL Period Number	Y	CAL_PERIOD_GL_PERIOD_NUM	This is the same value as CAL_PERIOD_NUMBER (one of the components in the Calendar Period identifier).
Calendar Period Source System Code	Y	SOURCE_SYSTEM_CODE	Identifies the originating source for the member.
Calendar Period Prefix	Y	CAL_PERIOD_PREFIX	This is a free-form alphanumeric attribute used as a prefix for the Calendar Period Name.

Company Cost Center Organization Dimension

Attributes	Required	Labels	Attribute Description
Hidden Flag	Y	HIDDEN_FLAG	Identifies dimension members that are used only for planning purposes and are not used outside of the Enterprise Performance Foundation applications. A 'Y' indicates such a member.
Reconciliation Child Only Flag	Y	RECON_LEAF_NODE_FLAG	Designates if the Company Cost Center Organization is included for Reconciliation purposes. If the value = 'Y', then the Company Cost Center Organization is a not a Parent and is therefore included in calculations for lowest level reconciliation. If a Company Cost Center Organization does not have an assigned value for this attribute, then it is assumed to be a Parent and is therefore not included in such calculations.
Company Cost Center Org Source System Code	Y	SOURCE_SYSTEM_CODE	Identifies the originating source for the member.

Attributes	Required	Labels	Attribute Description
Company Cost Center Organization Type	N	CCTR_ORG_TYPE	Allows for grouping of organizational units into functional groups for use in the selector, reports, and worksheets The list of values for this attribute comes from the Company Cost Center Organization Type dimension.
Company	N	COMPANY	Identifies the Company member from which the Company Cost Center Organization pair was created. The list of values for this attribute comes from the Company dimension.
Cost Center	N	COST_CENTER	Identifies the Cost Center member from which the Company Cost Center Organization pair was created. The list of values for these attributes comes from the Cost Center dimension.

Channel Dimension

Attribute	Required	Label	Attribute Description
Channel Source System Code	Y	SOURCE_SYSTEM_CODE	Identifies the originating source for the member.
Reconciliation Child Only Flag	Y	RECON_LEAF_NODE_FLAG	Designates if the member is included for Reconciliation purposes. If the value = 'Y', then the member is a not a Parent and is therefore included in calculations for lowest level reconciliation. If a member has a 'N' for this attribute, then it is assumed to be a Parent and is therefore not included in such calculations.

Customer Dimension

Attribute	Required	Label	Attribute Description
Reconciliation Child Only Flag	Y	RECON_LEAF_NODE_FLAG	Designates if the member is included for Reconciliation purposes. If the value = 'Y', then the member is a not a Parent and is therefore included in calculations for lowest level reconciliation. If a member has a 'N' for this attribute, then it is assumed to be a Parent and is therefore not included in such calculations.
Customer Source System Code	Y	SOURCE_SYSTEM_CODE	Identifies the originating source for the member.

Entity Dimension

Attribute	Required	Label	Attribute Description
Entity Source System Code	Y	SOURCE_SYSTEM_CODE	Identifies the originating source for the member.
Reconciliation Child Only Flag	Y	RECON_LEAF_NODE_FLAG	Designates if the member is included for Reconciliation purposes. If the value = 'Y', then the member is a not a Parent and is therefore included in calculations for lowest level reconciliation. If a member has a 'N' for this attribute, then it is assumed to be a Parent and is therefore not included in such calculations.
Entity Type Code	Y	ENTITY_TYPE_CODE	Differentiates the different categories of Entity dimension members. Comes from the <i>Entity Type Code</i> dimension. Possible Entity Types are: <ul style="list-style-type: none"> • 'C' – Consolidation • 'E' – Elimination • 'O' – Operating

Attribute	Required	Label	Attribute Description
Operating Entity ID	N	OPERATING_ENTITY_ID	For CONSOLIDATION Entities only – identifies the core operating Entity. The Operating Entity is the wholly owned subsidiary by the Consolidation Entity.
Elimination Entity ID	N	ELIMINATION_ENTITY_ID	For CONSOLIDATION Entities only - The Elimination Entity is the associated entity holding all the eliminations for a particular Consolidation Entity.
Image Name	N	IMAGE_NAME	The filename of a file located in the middle-tier under (OA_HTML/p) and used for viewing purposes in Advanced Global Consolidations User Interfaces.

Geography Dimension

Attribute	Required	Label	Attribute Description
Reconciliation Child Only Flag	Y	RECON_LEAF_NODE_FLAG	Designates if the member is included for Reconciliation purposes. If the value = 'Y', then the member is a not a Parent and is therefore included in calculations for lowest level reconciliation. If a member has a 'N' for this attribute, then it is assumed to be a Parent and is therefore not included in such calculations.
Geography Source System Code	Y	SOURCE_SYSTEM_CODE	Identifies the originating source for the member.

Line Item Dimension

Attribute	Required	Label	Attribute Description
Default Aggregation Method	N	DEFAULT_AGG_METHOD	Attribute for Analytic Workspace.
Budgeting Allowed Flag	Y	BUDGET_ALLOWED_FLAG	Specifies if Budgeting is allowed for the Line Item.
Hidden Flag	Y	HIDDEN_FLAG	Identifies dimension members that are used only for planning purposes and are not used outside of the Enterprise Performance Foundation applications. A 'Y' indicates such a member.
Account Type	Y	EXTENDED_ACCT_TYPE	Used to determine what financial elements are to be output in processes. Also used to determine signage of debits and credits written to Enterprise Performance Foundation.
Line ItemSource System Code	Y	SOURCE_SYSTEM_CODE	Identifies the originating source for the member.
Reconciliation Child Only Flag	Y	RECON_LEAF_NODE_FLAG	Designates if the Line Item is included for Reconciliation purposes. If the value for the flag code = 'Yes', then the Line Item is a not a Parent and is therefore included in calculations for lowest level reconciliation. If a Line Item does not have an assigned value for this attribute, then it is assumed to be a Parent and is therefore not included in such calculations.
Time Aggregation Method	N	TIME_AGG_METHOD	Attribute for Analytic Workspace.

Attribute	Required	Label	Attribute Description
Default Number Format	N	DEFAULT_NUMBER_FORMAT	Designates the Oracle Standard Query Language number format mask for displaying values of the line item.
Better Flag	Y	BETTER_FLAG	A 'Y' designates that the Line Item employs a Better indicator (+1), while an 'N' designates that the Line Item employs a Worse indicator

Natural Account Dimension

Attribute	Required	Label	Attribute Description
Extended Account Type	Y	EXTENDED_ACCT_TYPE	The CCE requires a mechanism to designate the sign (+ or -) for results at the Line Item and Natural Account dimensions. The account type attribute (Revenue, Expense, Asset, and so on) drives this mechanism.
Budgeting Allowed Flag	Y	BUDGET_ALLOWED_FLAG	Specifies if Budgeting is allowed for the Natural Account.
Expense Type Code	N	NAT_ACCT_EXPENSE_TYPE_CODE	Specifies the nature of expense accounts. Expense accounts are labeled as either 'FIXED' or 'VARIABLE'. If the account is not an expense account, then no attribute row exists for it.
Inventoriable Flag	Y	INVENTORIABLE_FLAG	Natural accounts may be indicated as inventoriable and non-inventoriable. This attribute is <i>inherited</i> throughout the costing process, and the inventoriable cost component for a cost object may be indicated.
Reconciliation Child Only Flag	Y	RECON_LEAF_NODE_FLAG	Designates if the member is included for Reconciliation purposes. If the value = 'Y', then the member is a not a Parent and is therefore included in calculations for lowest level reconciliation. If a member has a 'N' for this attribute, then it is assumed to be a Parent and is therefore not included in such calculations.
Natural Account Source System Code	Y	SOURCE_SYSTEM_CODE	Identifies the originating source for the member.

Project Dimension

The attributes for this dimension are not defined at this time.

Product Dimension

Attribute	Required	Label	Attribute Description
Unit of Measure	N	PRODUCT_UOM	This identifies the Unit of Measure attached to the Product member. When a Cost Object composite member (with Product as a component dimension) is created, this attribute assignment is used as the Unit of Measure for the Cost Object.
Material Flag	Y	PRODUCT_MATERIAL_FLAG	Identifies if the product is a material or a cost object. A 'Y' means that the product is a material. An 'N' does not necessarily mean the product is a cost object. A cost object may comprise additional dimensions and may not include product.
Reconciliation Child Only Flag	Y	RECON_LEAF_NODE_FLAG	Designates if the member is included for Reconciliation purposes. If the value = 'Y', then the member is a not a Parent and is therefore included in calculations for lowest level reconciliation. If a member has an 'N' for this attribute, then it is assumed to be a Parent and is therefore not included in such calculations.
Product Source System Code	Y	SOURCE_SYSTEM_CODE	Identifies the originating source for the member.

User-Configurable Dimensions (1-10)

Attribute	Required	Label	Attribute Description
User Dimx Reconciliation Child Only Flag	Y	RECON_LEAF_NODE_FLAG	Designates if the member is included for Reconciliation purposes. If the value = 'Y', then the member is a not a Parent and is therefore included in calculations for lowest level reconciliation. If a member has an 'N' for this attribute, then it is assumed to be a Parent and is therefore not included in such calculations.
User Dimx Source System Code	Y	SOURCE_SYSTEM_CODE	Identifies the originating source for the User-Configurable Dimension.

Table Descriptions

FEM_xName_B_T

This interface table stores dimension members for import into Enterprise Performance Foundation. Example names for these interface tables are as follows:

- FEM_PRODUCTS_B_T
- FEM_CHANNELS_B_T
- FEM_NAT_ACCTS_B_T

Column Name	Datatype	Nullable	Column Description
xName_DISPLAY_CODE	VARCHAR2(150)	Not Null	Alphanumeric code identifying the dimension member.
VALUE_SET_DISPLAY_CODE	VARCHAR2(150)	Not Null	Alphanumeric code of the Value Set for the member.
DIMENSION_GROUP_DISPLAY_CODE	VARCHAR2(80)		Identifies the Level (Dimension Group) for the member.
STATUS	VARCHAR2(30)	Not Null	Designates the Status of the row for loading. Records with a STATUS = 'LOAD' are loaded by the Enterprise Performance Foundation Dimension Member Loader. All other STATUS codes designate some sort of error from the load. Note that the <i>Reprocess Errors Flag</i> in FEM_DIM_LOAD_STATUS enables users to load records that have an error status without requiring them to update the STATUS column back to 'LOAD'.

The primary key is on the following columns:

- xName_DISPLAY_CODE
- VALUE_SET_DISPLAY_CODE

FEM_xName_TL_T

This interface table stores dimension member names and descriptions for import into Enterprise Performance Foundation. Example names for these interface tables are as follows:

- FEM_PRODUCTS_TL_T
- FEM_CHANNELS_TL_T
- FEM_NAT_ACCTS_TL_T

Column Name	Datatype	Nullable	Column Description
xName_DISPLAY_CODE	VARCHAR2(150)	Not Null	Alphanumeric code identifying the dimension member.
VALUE_SET_DISPLAY_CODE	VARCHAR2(150)	Not Null	Alphanumeric code of the Value Set for the member.
LANGUAGE	VARCHAR2(4)	Not Null	Identifies the language for the name/descriptions
XName_NAME	VARCHAR2(150)	Not Null	Translatable name for the member.
DESCRIPTION	VARCHAR2(255)	Not Null	Translatable description for the member.
STATUS	VARCHAR2(30)	Not Null	Designates the Status of the row for loading. Records with a STATUS ≠ 'LOAD' are loaded by the Enterprise Performance Foundation Dimension Member Loader. All other STATUS codes designate some sort of error from the load.

The primary key is on the following columns:

- xName_DISPLAY_CODE
- VALUE_SET_DISPLAY_CODE
- LANGUAGE

FEM_SIMPLE_DIMS_B_T

This interface table stores dimension members for import into Enterprise Performance Foundation.

Column Name	Datatype	Nullable	Column Description
DIMENSION_ VARCHAR_LABEL	VARCHAR2(30)	Not Null	Alphanumeric code identifying the dimension.
MEMBER_CODE	VARCHAR2(150)	Not Null	Alphanumeric code identifying the dimension member.
STATUS	VARCHAR2(30)	Not Null	Designates the Status of the row for loading. Records with a STATUS = 'LOAD' are loaded by the Enterprise Performance Foundation Dimension Member Loader. All other STATUS codes designate some sort of error from the load.

The primary key is on the following columns:

- DIMENSION_VARCHAR_LABEL
- MEMBER_CODE

FEM_SIMPLE_DIMS_TL_T

This interface table stores dimension member names and descriptions for the generic *simple* dimensions for import into Enterprise Performance Foundation. The generic simple dimensions are 200 VARCHAR dimensions that do not have hierarchies or attributes and do not require value set in the member identifier. These generic dimensions are available for use as lists of values dimensions.

This table supports data for multiple dimensions at a time because it has the DIMENSION_VARCHAR_LABEL column to differentiate the data by dimension.

Column Name	Datatype	Nullable	Column Description
DIMENSION_ VARCHAR_ LABEL	VARCHAR2(30)	Not Null	Alphanumeric code identifying the dimension.
MEMBER_DISPLAY_ CODE	VARCHAR2(150)	Not Null	Alphanumeric code identifying the dimension member.
LANGUAGE	VARCHAR2(4)	Not Null	Identifies the language for the name/ descriptions
MEMBER_NAME	VARCHAR2(150)	Not Null	Translatable name for the member.
DESCRIPTION	VARCHAR2(255)	Not Null	Translatable description for the member.
STATUS	VARCHAR2(50)	Not Null	Designates the Status of the row for loading. Records with a STATUS = 'LOAD' are loaded by the Enterprise Performance Foundation Dimension Member Loader. All other STATUS codes designate some sort of error from the load.

The primary key is on the following columns:

- DIMENSION_ VARCHAR_ LABEL
- MEMBER_DISPLAY_ CODE
- LANGUAGE

FEM_CAL_PERIODS_B_T

This interface table stores Calendar Period dimension members for import into Enterprise Performance Foundation.

Column Name	Datatype	Nullable	Column Description
CALENDAR_DISPLAY_CODE	VARCHAR2(150)	Not Null	Identifies the Calendar.
DIMENSION_GROUP_DISPLAY_CODE	VARCHAR2(80)	Not Null	Identifies the Level (Dimension Group) for the member.
CAL_PERIOD_END_DATE	DATE	Not Null	Ending date for the period.
CAL_PERIOD_NUMBER	NUMBER(15)	Not Null	This is the Period Number for the period. For example, for MONTH, January = 1, February = 2, March = 3, and so on. For Days, Jan-01 = 1, Jan-02 = 2, Jan-03=3, and so on. This number resets for each year.
STATUS	VARCHAR2(30)	Not Null	Designates the Status of the row for loading. Records with a STATUS = 'LOAD' are loaded by the Enterprise Performance Foundation Dimension Member Loader. All other STATUS codes designate some sort of error from the load.

The primary key is on the following columns:

- CALENDAR_DISPLAY_CODE
- DIMENSION_GROUP_DISPLAY_CODE
- CAL_PERIOD_END_DATE
- CAL_PERIOD_NUMBER

FEM_CAL_PERIODS_TL_T

This interface table stores Calendar Period dimension member names and descriptions for import into Enterprise Performance Foundation.

Column Name	Datatype	Nullable	Column Description
CALENDAR_DISPLAY_CODE	VARCHAR2(150)	Not Null	It identifies the Calendar.
DIMENSION_GROUP_DISPLAY_CODE	VARCHAR2(80)	Not Null	Identifies the Level (Dimension Group) for the member.
CAL_PERIOD_END_DATE	DATE	Not Null	Ending date for the period.
CAL_PERIOD_NUMBER	NUMBER(15)	Not Null	This is the Period Number for the period. For example, for MONTH, January = 1, February = 2, March = 3, and so on. For Days, Jan-01 = 1, Jan-02 = 2, Jan-03=3, and so on. This number resets for each year.
LANGUAGE	VARCHAR2(4)	Not Null	Identifies the language for the name/ descriptions and Calendar.
CAL_PERIOD_NAME	VARCHAR2(150)	Not Null	Translatable name for the member.
DESCRIPTION	VARCHAR2(255)	Not Null	Translatable description for the member.
STATUS	VARCHAR2(30)	Not Null	Designates the Status of the row for loading. Records with a STATUS = 'LOAD' are loaded by the Enterprise Performance Foundation Dimension Member Loader. All other STATUS codes designate some sort of error from the load.

The primary key is on the following columns:

- CALENDAR_DISPLAY_CODE
- DIMENSION_GROUP_DISPLAY_NAME
- CAL_PERIOD_END_DATE
- CAL_PERIOD_NUMBER
- LANGUAGE

FEM_xName_ATTR_T

This interface table stores dimension attributes for import into the Enterprise Performance Foundation database. Attributes are identified by the ATTRIBUTE_VARCHAR_LABEL as specified in the *Dimension: Attributes* page. You need the information in the Code Column.

The VERSION_DISPLAY_CODE is required for all attribute assignments. If the VERSION_DISPLAY_CODE (for the designated LANGUAGE) already exists in Enterprise Performance Foundation, then the attribute assignments *overwrite* the existing attribute assignments for that version. If the VERSION_DISPLAY_CODE does not exist, the loader does not load the assignment and updates the STATUS of the record appropriately.

Example names for these interface tables are as follows:

- FEM_PRODUCTS_ATTR_T
- FEM_CHANNELS_ATTR_T
- FEM_NAT_ACCTS_ATTR_T

Column Name	Datatype	Nullable	Column Description
xName_DISPLAY_CODE	VARCHAR2(150)	Not Null	Alphanumeric code identifying the dimension member.
VALUE_SET_DISPLAY_CODE	VARCHAR2(150)	Not Null	Alphanumeric code of the Value Set for the member.
ATTRIBUTE_VARCHAR_LABEL	VARCHAR2(30)	Not Null	Identifies the Attribute. The Enterprise Performance Foundation Dimension Member Loader uses this value to look up the ATTRIBUTE_ID in FEM_DIM_ATTRIBUTES_B.

Column Name	Datatype	Nullable	Column Description
ATTRIBUTE_ASSIGN_VALUE	VARCHAR2(1000)	Not Null	<p>This is the attribute assignment. Attribute assignments of all types are stored in this column – NUMBER, VARCHAR2, and DATE. The Enterprise Performance Foundation Dimension Member Loader converts the attribute assignment to the proper data type and stores it in the appropriate column of the ATTR table.</p> <p>DATE values must be stored in the format as designated by the date format mask input parameter of the Dimension Member Loader. Value for DATE attributes that are not in this format are flagged as error rows.</p>
VERSION_DISPLAY_CODE	VARCHAR2(80)	Not Null	<p>The VERSION_DISPLAY_CODE identifies the VERSION_ID for the attribute assignment. If no existing Version matches for the attribute, the Enterprise Performance Foundation Dimension Member Loader returns with a STATUS error.</p>
ATTR_ASSIGN_VALUE_DISPLAY_CODE	VARCHAR2(150)		<p>If the Attribute assignment is a Value Set Required (VSR) dimension member, this column designates the value set for the member.</p>

Column Name	Datatype	Nullable	Column Description
CALPATTR_CAL_D ISPLAY_CODE	VARCHAR2(150)	Null	Non translatable alphanumeric identifier of the Calendar that is a component of the attribute assignment when the attribute references the Calendar Period dimension as a list of values
CALPATTR_DIMGR P_DISPLAY_CODE	VARCHAR2(150)	Null	Non translatable alphanumeric identifier of the Dimension Group that is a component of the attribute assignment when the attribute references the Calendar Period dimension as a list of values
CALPATTR_END_ DATE	DATE	Null	End Date of the Calendar Period that is a component of the attribute assignment when the attribute references the Calendar Period dimension as a list of values
CALPATTR_PER IOD_NUM	NUMBER	Null	Period Number of the Calendar Period that is a component of the attribute assignment when the attribute references the Calendar Period dimension as a list of values

Column Name	Datatype	Nullable	Column Description
CREATED_BY_DIM_MIGRATION_FLAG	VARCHAR2(1)	Null	Identifies rows created by the dimension migration process
STATUS	VARCHAR2(30)	Not Null	Designates the Status of the row for loading. Records with a STATUS = 'LOAD' are loaded by the Enterprise Performance Foundation Dimension Member Loader. All other STATUS codes designate some sort of error from the load.

The primary key is on the following columns:

- xName_DISPLAY_CODE
- VALUE_SET_DISPLAY_CODE
- ATTRIBUTE_VARCHAR_LABEL
- ATTRIBUTE_ASSIGN_VALUE
- PRODUCT_DISPLAY_CODE
- VERSION_DISPLAY_CODE
- ATTR_ASSIGN_VS_DISPLAY_CODE
- CALPATTR_CAL_DISPLAY_CODE
- CALPATTR_DIMGRP_DISPLAY_CODE
- CALPATTR_END_DATE
- CALPATTR_PERIOD_NUM

FEM_CAL_PERIODS_ATTR_T

This interface table stores Calendar Period dimension attributes for import into the Enterprise Performance Foundation database. Attributes are identified by the ATTRIBUTE_VARCHAR_LABEL as specified in the *Dimension: Attributes* page. You need the information in the Code Column.

Note: The VERSION_NAME is required for all attribute assignments. If the VERSION_NAME (for the designated LANGUAGE) already exists in FEM, then the attribute assignments *overwrite* the existing attribute assignments for that version. If the VERSION_NAME does not exist yet, then a new version is created.

Column Name	Datatype	Nullable	Column Description
CALENDAR_DISPLAY_CODE	VARCHAR2(150)	Not Null	Identifies the Calendar.
DIMENSION_GROUP_DISPLAY_CODE	VARCHAR2(80)	Not Null	Identifies the Level (Dimension Group) for the member.
CAL_PERIOD_END_DATE	DATE	Not Null	Ending date for the period.
CAL_PERIOD_NUMBER	NUMBER(15)	Not Null	This is the Period Number for the period. For example, for MONTH, January = 1, February = 2, March = 3 and so on. For Days, Jan-01 = 1, Jan-02 = 2, Jan-03=3, and so on. This number resets for each year.
ATTRIBUTE_VARIABLE_LABEL	VARCHAR2(30)	Not Null	Identifies the Attribute. The Enterprise Performance Foundation Dimension Member Loader uses this value to look up the ATTRIBUTE_ID in FEM_DIM_ATTRIBUTES_B.
ATTRIBUTE_ASSIGNMENT_VALUE	VARCHAR2(1000)	Not Null	This is the attribute assignment. Attribute assignments of all types are stored in this column – NUMBER, VARCHAR2, and DATE. The Enterprise Performance Foundation Dimension Member Loader converts the attribute assignment to the proper data type and stores it in the appropriate column of the ATTR table. DATE values must be stored in the format as designated by the date format mask input parameter of the Dimension Member Loader. Values for DATE attributes that are not in this format are flagged as error rows.

Column Name	Datatype	Nullable	Column Description
VERSION_DISPLAY_CODE	VARCHAR2(80)	Not Null	The VERSION_DISPLAY_CODE identifies the VERSION_ID for the attribute assignment. If no existing Version matches for the attribute, the Enterprise Performance Foundation Dimension Member Loader returns with a STATUS error.
ATTR_ASSIGN_VS_DISPLAY_CODE	VARCHAR2(150)		If the Attribute assignment is a Value Set Required (VSR) dimension member, this column designates the value set for the member.
STATUS	VARCHAR2(30)	Not Null	Designates the Status of the row for loading. Records with a STATUS = 'LOAD' are loaded by the Enterprise Performance Foundation Dimension Member Loader. All other STATUS codes designate some sort of error from the load.

The primary key is on the following columns:

- CALENDAR_DISPLAY_CODE
- DIMENSION_GROUP_DISPLAY_CODE
- CAL_PERIOD_END_DATE
- CAL_PERIOD_NUMBER
- ATTRIBUTE_VARCHAR_LABEL
- ATTRIBUTE_ASSIGN_VALUE

FEM_DIMENSION_GRP_B_T

This interface table stores the Levels (Dimension Groups) for loading. It is populated only when the user has new levels to be loaded into Enterprise Performance Foundation.

Column Name	Datatype	Nullable	Column Description
DIMENSION_GROUP_DISPLAY_CODE	VARCHAR2(80)	Not Null	Identifies the Level (Dimension Group) for loading.
DIMENSION_VARCHAR_LABEL	VARCHAR2(30)	Not Null	Identifies the Dimension of the Level.
DIMENSION_GROUP_SEQ	NUMBER(9)	Not Null	Designates the relative hierarchical relationship of the Level within the Dimension.
STATUS	VARCHAR2(30)	Not Null	Designates the Status of the row for loading. Records with a STATUS = 'LOAD' are loaded by the Enterprise Performance Foundation Dimension Member Loader. All other STATUS codes designate some sort of error from the load.

The primary key is on the following columns:

- DIMENSION_GROUP_DISPLAY_CODE
- DIMENSION_VARCHAR_LABEL

FEM_DIMENSION_GRPS_TL_T

This interface table stores the translatable names/descriptions for the loading of Levels (Dimension Groups). It is populated only when the user has new Levels to be loaded into Enterprise Performance Foundation. The Dimension Member Loader supports updating the names/descriptions of existing Levels using this interface table.

Column Name	Datatype	Nullable	Column Description
DIMENSION_GROUP_DISPLAY_CODE	VARCHAR2(150)	Not Null	Identifies the Dimension Group.
DIMENSION_VARCHAR_LABEL	VARCHAR2(30)	Not Null	Identifies the Dimension of the Group.
DIMENSION_GROUP_NAME	VARCHAR2(80)	Not Null	Translatable name for the Dimension Group.
LANGUAGE	VARCHAR2(4)	Not Null	Designates the Language of the DIMENSION_GROUP_NAME.
DESCRIPTION	VARCHAR2(255)		Description for the Dimension Group.
STATUS	VARCHAR2(30)	Not Null	Designates the Status of the row for loading. Records with a STATUS = 'LOAD' are loaded by the Enterprise Performance Foundation Dimension Member Loader. All other STATUS codes designate some sort of error from the load.

The primary key is on the following columns:

- DIMENSION_GROUP_DISPLAY_CODE
- DIMENSION_VARCHAR_LABEL

There is an alternate unique index on the following:

- DIMENSION_GROUP_NAME
- DIMENSION_VARCHAR_LABEL
- LANGUAGE

Sample SQL*Loader DAT and CTL Files for the Dimension Member Interface Tables

The following section provides examples of SQL*Loader DAT and CTL files for loading into the Dimension Member interface tables.

Levels Base Interface Table

- CTL file for FEM_DIMENSION_GRP_B_T

```

OPTIONS (ERRORS=0, SILENT=(FEEDBACK), DIRECT=TRUE)
LOAD DATA
INFILE 'dimension_grps_b_t.dat'
BADFILE 'dimension_grps_b_t.bad'
DISCARDFILE 'dimension_grps_b_t.dsc'
INTO TABLE FEM_DIMENSION_GRPS_B_T
REPLACE
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
(
  DIMENSION_GROUP_DISPLAY_CODE,
  DIMENSION_VARCHAR_LABEL,
  DIMENSION_GROUP_SEQ,
  TIME_GROUP_TYPE_CODE,
  STATUS
)

```

- **DAT file for FEM_DIMENSION_GRPS_B_T**

```

"CAL_PERIOD_L1","CAL_PERIOD","1","YEAR","LOAD"
"LINE_ITEM_L4","LINE_ITEM","4","","LOAD"

```

- **CTL file for FEM_DIMENSION_GRPS_TL_T**

```

OPTIONS (ERRORS=0, SILENT=(FEEDBACK), DIRECT=TRUE)
LOAD DATA
INFILE 'dimension_grps_tl_t.dat'
BADFILE 'dimension_grps_tl_t.bad'
DISCARDFILE 'dimension_grps_tl_t.dsc'
INTO TABLE FEM_DIMENSION_GRPS_TL_T
REPLACE
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
(
  DIMENSION_GROUP_DISPLAY_CODE,
  LANGUAGE,
  DIMENSION_GROUP_NAME,
  DESCRIPTION,
  STATUS,
  DIMENSION_VARCHAR_LABEL
)

```

- **DAT file for FEM_DIMENSION_GRPS_TL_T**

```

"CAL_PERIOD_L1","US","Time Level L1","Time Level L1","LOAD",
"CAL_PERIOD","LINE_ITEM_L4","US","Line Item Level L4",
"Line Item Level L4","LOAD","LINE_ITEM"

```

Line Item Base Interface Table

- **CTL file for FEM_LN_ITEMS_B_T**

```

OPTIONS (ERRORS=0, SILENT=(FEEDBACK), DIRECT=TRUE)
LOAD DATA
INFILE 'line_item_b_t.dat'
BADFILE 'line_item_b_t.bad'
DISCARDFILE 'line_item_b_t.dsc'
INTO TABLE FEM_LN_ITEMS_B_T
REPLACE
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
(
LINE_ITEM_DISPLAY_CODE,
VALUE_SET_DISPLAY_CODE,
DIMENSION_GROUP_DISPLAY_CODE,
STATUS
)

```

- **DAT file for FEM_LN_ITEMS_B_T**

```

"LN0001","LINE_ITEM","LINE_ITEM_L4","LOAD"
"LN0026","LINE_ITEM","LINE_ITEM_L4","LOAD"

```

Line Item TL Interface Table

- **CTL file for FEM_LN_ITEMS_TL_T**

```

OPTIONS (ERRORS=0, SILENT=(FEEDBACK), DIRECT=TRUE)
LOAD DATA
INFILE 'line_item_tl_t.dat'
BADFILE 'line_item_tl_t.bad'
DISCARDFILE 'line_item_tl_t.dsc'
INTO TABLE FEM_LN_ITEMS_TL_T
REPLACE
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
(
LINE_ITEM_DISPLAY_CODE,
VALUE_SET_DISPLAY_CODE,
LANGUAGE,
LINE_ITEM_NAME,
DESCRIPTION,
STATUS
)

```

- **DAT file for FEM_LN_ITEMS_TL_T**

```

"LN0001","LINE_ITEM","US","All Drivers","All Drivers", "LOAD","LN
0026","LINE_ITEM","US","Resource Related Lines",
"Resource Related Lines","LOAD"

```

Line Item ATTR Interface Table

- **CTL file for FEM_LN_ITEMS_ATTR_T**


```

OPTIONS (ERRORS=0, SILENT=(FEEDBACK), DIRECT=TRUE)
LOAD DATA
INFILE 'line_item_attr_t.dat'
BADFILE 'line_item_attr_t.bad'
DISCARDFILE 'line_item_attr_t.dsc'
INTO TABLE FEM_LN_ITEMS_ATTR_T
REPLACE
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
(
LINE_ITEM_DISPLAY_CODE,
VALUE_SET_DISPLAY_CODE,
ATTRIBUTE_VARCHAR_LABEL,
ATTRIBUTE_ASSIGN_VALUE,
VERSION_DISPLAY_CODE,
STATUS
)

```

- **DAT file for FEM_LN_ITEMS_ATTR_T**

```

"LN0001","LINE_ITEM","ACTIVATION_FLAG","N","Default","LOAD"
"LN0002","LINE_ITEM","ACTIVATION_FLAG","N","Default","LOAD"

```

Calendar Period Base Interface Table

- **CTL file for FEM_CAL_PERIODS_B_T**

```

OPTIONS (ERRORS=0, SILENT=(FEEDBACK), DIRECT=TRUE)
LOAD DATA
INFILE 'cal_period_b_t.dat'
BADFILE 'cal_period_b_t.bad'
DISCARDFILE 'cal_period_b_t.dsc'
INTO TABLE FEM_CAL_PERIODS_B_T
REPLACE
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
(
CAL_PERIOD_END_DATE,
CAL_PERIOD_NUMBER,
DIMENSION_GROUP_DISPLAY_CODE,
STATUS,
CALENDAR_DISPLAY_CODE
)

```

- **DAT file for FEM_CAL_PERIODS_B_T**

```

"31-DEC-2004","1","CAL_PERIOD_L1","LOAD","Standard"
"31-DEC-2005","1","CAL_PERIOD_L1","LOAD","Standard"

```

Calendar Period TL Interface Table

- **CTL file for FEM_CAL_PERIODS_TL_T**

```

OPTIONS (ERRORS=0, SILENT=(FEEDBACK), DIRECT=TRUE)
LOAD DATA
INFILE 'cal_period_tl_t.dat'
BADFILE 'cal_period_tl_t.bad'
DISCARDFILE 'cal_period_tl_t.dsc'
INTO TABLE FEM_CAL_PERIODS_TL_T
REPLACE
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
(
CAL_PERIOD_END_DATE,
CAL_PERIOD_NUMBER,
DIMENSION_GROUP_DISPLAY_CODE,
CALENDAR_DISPLAY_CODE,
LANGUAGE,
CAL_PERIOD_NAME,
DESCRIPTION,
STATUS
)

```

- **DAT file for FEM_CAL_PERIODS_TL_T**

```

"31-DEC-2004","1","CAL_PERIOD_L1","Standard","US","Year 2004",
"Year 2004","LOAD"
"31-DEC-2005","1","CAL_PERIOD_L1","Standard","US","Year 2005",
"Year 2005","LOAD"

```

Calendar Period ATTR Interface Table

- **CTL file for FEM_CAL_PERIODS_ATTR_T**

```

OPTIONS (ERRORS=0, SILENT=(FEEDBACK), DIRECT=TRUE)
LOAD DATA
INFILE 'cal_period_attr_t.dat'
BADFILE 'cal_period_attr_t.bad'
DISCARDFILE 'cal_period_attr_t.dsc'
INTO TABLE FEM_CAL_PERIODS_ATTR_T
REPLACE
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
(
CALENDAR_DISPLAY_CODE,
CAL_PERIOD_END_DATE,
CAL_PERIOD_NUMBER,
DIMENSION_GROUP_DISPLAY_CODE,
ATTRIBUTE_VARCHAR_LABEL,
ATTRIBUTE_ASSIGN_VALUE,
VERSION_DISPLAY_CODE,
STATUS
)

```

- **DAT file for FEM_CAL_PERIODS_ATTR_T**

```

"Standard","31-JAN-2004","1","CAL_PERIOD_L3",
"ACCOUNTING_YEAR","0","Default","LOAD"
"Standard","29-FEB-2004","2","CAL_PERIOD_L3",
"ACCOUNTING_YEAR","0","Default","LOAD"

```

Fact Interface Table Dimension Member Error Report

This program creates a report listing invalid display codes in dimension columns in any detail client data interface tables. It does not modify any data in the interface tables.

The details of the Fact Interface Table Dimension Member Error Report are as follows:

- Available as a concurrent program in the Enterprise Performance Foundation Administrator responsibility. This provides the ability to report on invalid entries in dimension columns in any interface table supported by the Detail Client Data loader. It will report on a maximum of 80 dimension columns per table (no current pre-defined client data tables exceed this limit).
- The report will display multiple dimensional errors in any given interface record. It provides an identifier (ID_Number if available, otherwise a database Row_ID), the dimension column name, and the first 80 characters of the invalid display code entry.
- The user may specify the maximum number of rows to validate (500 is the default).
- The user may also specify the maximum number of records to print (500 is the default). This refers to the number of problematic interface table rows it encounters, not the total number of errors. For example, if 5 rows have 3 errors each, it is considered 5 records, not 15 records.
- Output: It writes an optional report to the Output log; it does not modify any data in the interface tables.
- Similar to the Detail Client Data Loader itself, the user running the new Fact Interface Table Dimension Member Error Report engine requires at least read-only access to the Data Integration folder.

Running the Interface Dimension Error Report

1. From the Enterprise Performance Foundation Administrator responsibility, select the Fact Interface Table Dimension Member Error Report concurrent program.
2. Enter the parameters, including:
 - Associated loader rule (e.g., "Load data into Mortgages)
 - Set of data to validate, identified by: Ledger, Calendar Period, Dataset, and Source System
 - Maximum number of records to validate (defaults to 500 if left blank)
 - Maximum number of problematic records to print (defaults to 500 if left blank)
3. Print Report = **Yes**
4. To view your report, navigate through Process Management View Requests to find your Request ID, then check the Output for your report.

After you have run the program, you can also view the errors via a direct SQL query. You can obtain the SQL from the process log (navigate through Process Management to find your Request ID, then go through Details->View Log. The errors are stored in FEM_Interface_Fact_Errs, identified by Request ID. The rows are kept here indefinitely until you manually remove them from the FEM_Interface_Fact_Errs table.

Dimension Hierarchy Loader

Use the Dimension Hierarchy Loader to load hierarchy definitions. The Dimension Hierarchy Loader program supports the following with regard to hierarchies:

- Create new hierarchy objects

If the hierarchy object name does not yet exist in Enterprise Performance Foundation, the Dimension Hierarchy Loader creates a new hierarchy definition. It then loads the parent/child relationships from the interface table.

- Overwrite existing hierarchy definitions

If the hierarchy object name and the hierarchy definition exist, the Enterprise Performance Foundation External Loader program overwrites the existing parent/child relationships with new ones from the interface table. The Enterprise Performance Foundation External Loader program supports only the complete replacement of the parent/child relationships. It is not possible for the user to try to import just a subset of the parent/child relationships. Rather, all of the parent/child relationships are replaced with the parent/child relationships from the interface table.

Note: When you update parent/child relationships, you cannot change details in the FEM_HIERARCHIES_T existing row. If you do, they are ignored.

Warning: The Dimension Hierarchy Loader does not support loading hierarchies where any of the members are Personal members. If a hierarchy contains any such member in the interface tables, the load fails. See the *Enterprise Planning and Budgeting User's Guide* for more information about Personal members.

Note: You cannot reload a hierarchy version if that version has been process locked.

Hierarchy Dimensions

The Dimension Hierarchy Loader supports the loading of hierarchies for a pre-defined set of dimensions. It is not possible to load hierarchies for any other dimensions using this loader. As the structure of each of these dimensions varies, they can be separated into the following categories:

Time Dimension

- Calendar Period

Note: Calendar Period hierarchies are level-based.

Value Set Dimensions

- Channel
- Company Cost Center Organization
- Customer
- Entity
- Geography

- Line Item
- Natural Account
- Product
- Task
- Project
- Ten User-Configurable Hierarchical Dimensions

Time Dimension

The Calendar Period dimension is the only dimension in this category.

Identifying Calendar Period Members

Calendar Period is the only dimension whose identifier is a combination of the four separate components:

- Calendar identifier
- Level identifier
- End date
- Period number

It is the combination of the four components that guarantees uniqueness for each Calendar Period member within the interface tables.

When specifying parent/child relationships for Calendar Period hierarchies, use the following columns to identify individual Calendar Period members:

Column	Description
CALENDAR_DISPLAY_CODE (Calendar)	Each Calendar Period member belongs to one and only one Calendar. Typically, an installation has only a single Calendar defined for the entire system. In larger implementations, however, it is possible to have multiple Calendars defined. Therefore, you always need to provide a Calendar when identifying members of the Calendar Period dimension.
PARENT_DIM_GRP_DISPLAY_CODE and CHILD_DIM_GRP_DISPLAY_CODE (Level)	This is the Level to which the Calendar Period belongs. For example, Calendar Periods spanning 28, 30, and 31 days would be designated as belonging to the <i>Month</i> level. Calendar Periods spanning 24 hours would be designated as belonging to the <i>Day</i> level.
PARENT_CAL_PERIOD_END_DATE and CHILD_CAL_PERIOD_END_DATE (End Date)	This is the end date of the calendar period. For example, to designate the <i>March</i> period for the year 2004, the End Date is '31-MAR-2004'.
PARENT_CAL_PERIOD_NUMBER and CHILD_CAL_PERIOD_NUMBER (Period Number)	This is a relative number of the Calendar Period within a given Fiscal Year. The Period Number is a unique number that designates the sequential order of the Calendar Periods for periods of the same Level.

For example, if a company's Fiscal Year starts in June and ends in May the following year, and every period starts on the first of the month and ends on the last day of the month, there would be the following relationships among the Fiscal Year, Level (Dimension Group Display Code), End Date, and Period Number:

Module Fiscal Year	Period Name	Level (Dimension Group)	End Date	Period Number
2004	May-FY04	Month	31-May-2004	12
2005	Jun-FY05	Month	30-Jun-2004	1
2005	Jul-FY05	Month	31-Jul-2004	2
2005	Aug-FY05	Month	31-Aug-2004	3
2005	Sep-FY05	Month	30-Sep-2004	4
2005	Oct-FY05	Month	31-Oct-2004	5
2005	Nov-FY05	Month	30-Nov-2004	6
2005	Dec-FY05	Month	31-Dec-2004	7
2005	Jan-FY05	Month	31-Jan-2005	8
2005	Feb-FY05	Month	28-Feb-2005	9
2005	Mar-FY05	Month	31-Mar-2005	10
2005	Apr-FY05	Month	30-Apr-2005	11
2005	May-FY05	Month	31-May-2005	12
2005	Qtr1-FY05	Quarter	31-Aug-2004	1
2005	Qtr2-FY05	Quarter	30-Nov-2004	2
2005	Qtr3-FY05	Quarter	28-Feb-2005	3
2005	Qtr4-FY05	Quarter	31-May-2005	4
2005	Year-FY05	Year	31-May-2005	1
2006	Jun-FY06	Month	30-Jun-2005	1

Value Set Dimensions

A Value Set dimension is a dimension whose identifier includes Value Set as a component.

Identifying Value Set Dimension Members

You identify members for these dimensions using both an alphanumeric code for the member (also known as the *Member Display Code*) in combination with an alphanumeric code for the Value Set (also known as the *Value Set Display Code*). For example, the identifier for the Product dimension is the following columns:

- PRODUCT_DISPLAY_CODE
- VALUE_SET_DISPLAY_CODE

Typically there is only a single Value Set for all Product members. It is possible, however, for large implementations to employ multiple value sets for a dimension. Therefore, you must always provide the VALUE_SET_DISPLAY_CODE to uniquely identify a member of these dimensions.

Populating the Interface Tables for the Dimension Hierarchy Loader

The following section provides information on how to populate the Dimension Hierarchy Loader interface tables in preparation for a load.

You need to populate the appropriate set of interface tables to load hierarchies into Enterprise Performance Foundation. The interface tables that you populate depend upon the dimension that you are loading and the type of load that you are performing.

Hierarchy Definition Tables

Populate the following tables for hierarchy *definition* information. These tables store information about the type of hierarchy, as well as what levels and value sets may participate:

- FEM_HIERARCHIES_T
- FEM_HIER_DIM_GRPS_T
- FEM_HIER_VALUE_SETS_T

For any hierarchy load, whether you are creating a new hierarchy definition or updating an existing definition, you need to populate the FEM_HIERARCHIES_T table to identify the hierarchy definition to be loaded. For an update, all columns other than the identifying name columns are ignored.

If you are creating a new hierarchy definition (either for a new Hierarchy Object or for an existing Hierarchy Object), you also need to populate the FEM_HIER_DIM_GRPS_T and FEM_HIER_VALUE_SETS_T tables. These tables define the Levels and Value Sets that control the set of dimension members that may participate in the hierarchy. Do not populate either FEM_HIER_VALUE_SETS_T or FEM_HIER_DIM_GRPS_T when you are updating (that is, replacing) an existing hierarchy definition.

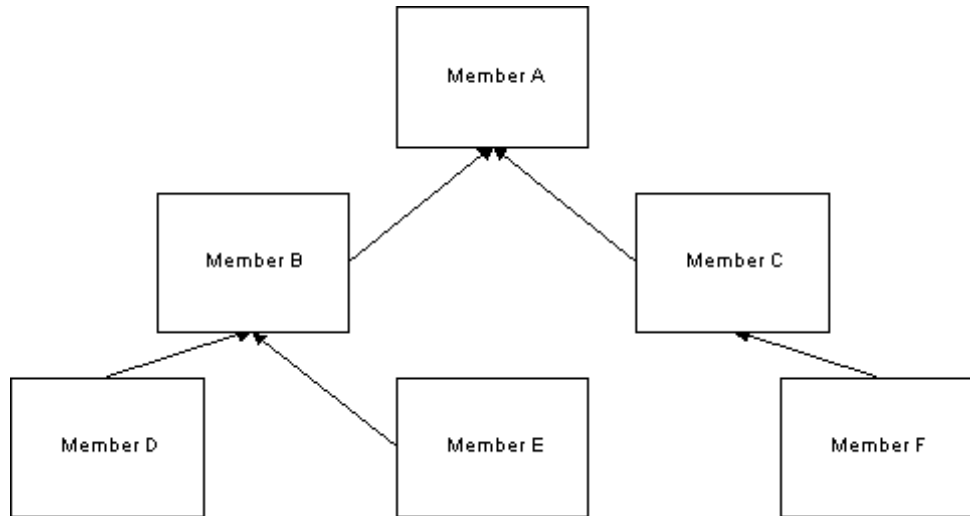
Parent/Child Relationship Tables

All hierarchy loads require population of the parent/child relationship table for the dimension being loaded. The following matrix lists these tables for each dimension:

Dimension	Parent/Child Table
Calendar Period	FEM_CAL_PERIODS_HIER_T
Channel	FEM_CHANNELS_HIER_T
Company Cost Center Organization	FEM_CCTR_ORGS_HIER_T
Customer	FEM_CUSTOMERS_HIER_T
Entity	FEM_ENTITIES_HIER_T
Geography	FEM_GEOGRAPHY_HIER_T
Line Item	FEM_LN_ITEMS_HIER_T
Natural Account	FEM_NAT_ACCTS_HIER_T
Product	FEM_PRODUCTS_HIER_T
Project	FEM_PROJECTS_HIER_T
Task	FEM_TASKS_HIER_T
User Dim1	FEM_USER_DIM1_HIER_T
User Dim2	FEM_USER_DIM2_HIER_T
User Dim3	FEM_USER_DIM3_HIER_T
User Dim4	FEM_USER_DIM4_HIER_T
User Dim5	FEM_USER_DIM5_HIER_T
User Dim6	FEM_USER_DIM6_HIER_T
User Dim7	FEM_USER_DIM7_HIER_T
User Dim8	FEM_USER_DIM8_HIER_T
User Dim9	FEM_USER_DIM9_HIER_T
User Dim10	FEM_USER_DIM10_HIER_T

You populate each HIER table with rows for each parent/child relationship in the Hierarchy definition. In addition, the loader requires that you provide a record identifying all root nodes of the hierarchy.

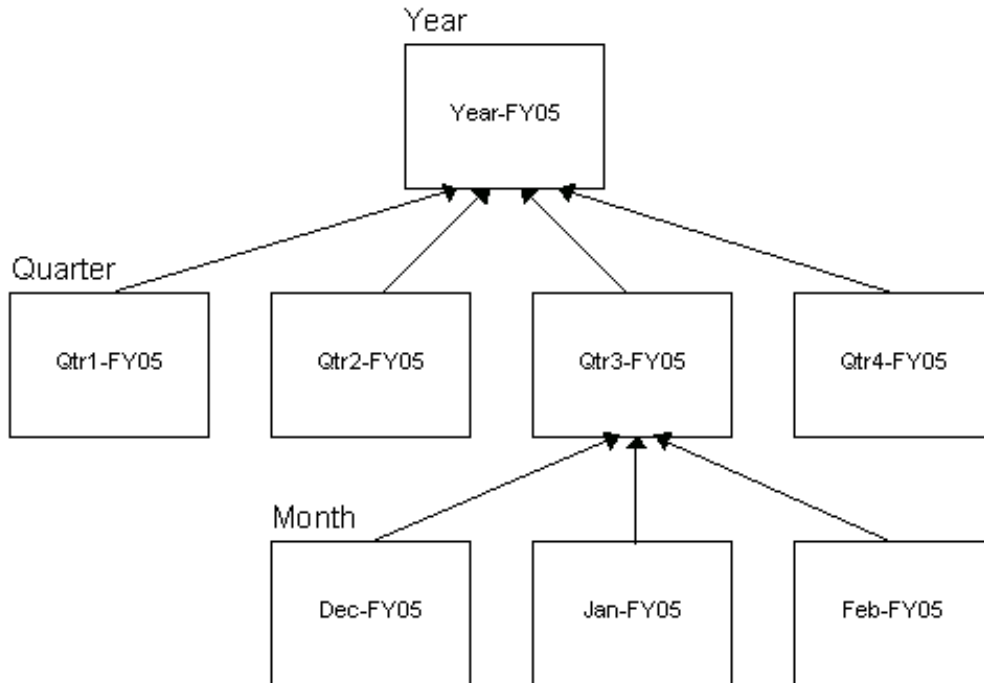
For example, assume the following hierarchy definition for a value set dimension:



The parent/child rows for the hierarchy would appear as follows in the `_HIER_T` table:

Parent Value Set	Parent Display Codes	Child Value Set	Child Display Code
123	A	123	A
123	A	123	B
123	A	123	C
123	B	123	D
123	B	123	E
123	C	123	F

The same hierarchy for the Calendar Period dimension appears as follows:



Representing hierarchies for Calendar Period dimension in the parent/child relationship table is different only in that the member identification employs four columns. For example:

Calendar Display Code	Parent Dimension Group	Parent End Date	Parent Period Number	Child Dimension Group	Child End Date	Child Period Number
Fiscal_Year	Year	31-May-2005	1	Year	31-May-2005	1
Fiscal_Year	Year	31-May-2005	1	Quarter	31-Aug-2004	1
Fiscal_Year	Year	31-May-2005	1	Quarter	30-Nov-2004	2
Fiscal_Year	Year	31-May-2005	1	Quarter	28-Feb-2005	3
Fiscal_Year	Year	31-May-2005	1	Quarter	31-May-2005	4
Fiscal_Year	Quarter	28-Feb-2005	3	Month	31-Dec-2004	7
Fiscal_Year	Quarter	28-Feb-2005	3	Month	31-Jan-2005	8
Fiscal_Year	Quarter	28-Feb-2005	3	Month	28-Feb-2005	9

Loader Operations for the Dimension Hierarchy Loader

This section describes the various types of loads that you can perform with the Dimension Hierarchy Loader.

Note: The Dimension Hierarchy Loader loads only a single hierarchy definition per run.

Create New Hierarchy Objects

The loader enables you to create a new hierarchy definition for a completely new hierarchy object. Populate the following tables for such a load:

- FEM_HIERARCHIES_T
Identify the new Hierarchy Object and Definition
- FEM_HIER_DIM_GRPS_T
Designate the Levels for the Hierarchy Object
- FEM_HIER_VALUE_SETS_T
Designate the Value Sets for the Hierarchy Object
- FEM_xName_HIER_T
Specify the parent/child relationships for the Hierarchy Definition

The loader checks the Hierarchy Object Name designated for the load. If the Hierarchy Object Name does not exist in the Enterprise Performance Foundation for the LANGUAGE (in FEM_HIERARCHIES_T), then the loader creates a new Hierarchy Object and new Hierarchy Object Definition.

Create New Hierarchy Definition for an Existing Hierarchy Object

The loader enables you to create a new hierarchy definition for an existing hierarchy object. Populate the following tables for such a load:

- FEM_HIERARCHIES_T (identify the existing Hierarchy Object and new Definition)
- FEM_xName_HIER_T (specify the parent/child relationships for the Hierarchy Definition)

The loader checks the Hierarchy Object Name designated for the load. If the Hierarchy Object Name exists in Enterprise Performance Foundation for the LANGUAGE (in FEM_HIERARCHIES_T), then the loader also checks the specified Hierarchy Definition Name. If the Hierarchy Definition Name does not exist for the designated Hierarchy Object Name, then the loader creates a new Hierarchy Object Definition for that specific Hierarchy Object.

Replace Hierarchy Definition

The loader enables you to replace the parent/child relationships for an existing hierarchy definition. Populate the following tables for such a load:

- FEM_HIERARCHIES_T (identify the existing Hierarchy Object and existing Definition)
- FEM_xName_HIER_T (specify the parent/child relationships for the Hierarchy Definition)

The loader checks the Hierarchy Object Name designated for the load. If the Hierarchy Object Name exists in Enterprise Performance Foundation for the LANGUAGE (in FEM_HIERARCHIES_T), then the loader also checks the specified Hierarchy Definition Name. If the Hierarchy Definition Name exists for the designated Hierarchy Object Name, then the loader replaces the parent/child relationship for that Definition Name.

Running the Dimension Hierarchy Loader

This section describes how to launch the loader from Concurrent Manager.

Concurrent Program

Choose the *Dimension Hierarchy Loader* concurrent program.

Restrictions

The Dimension Hierarchy Loader does not allow multiple users to load hierarchy definitions for the same hierarchy object at the same time. Users can simultaneously run the loader, however, as long as they are loading definitions for separate hierarchy objects.

Executed Rules

You cannot use the loader to remove any hierarchy definitions. You can update only the parent/child relationships for loaded hierarchy definitions.

If you need to permanently remove a hierarchy definition, you must use the Dimension and Hierarchy Management interface to perform such deletes.

Input Parameters

The Dimension Hierarchy Loader requires the following input parameters:

- Dimension
Pick a dimension from the pick list. This restricts the list of possible hierarchy objects for loading to only those of the designated dimension.
- Hierarchy Object
Pick a hierarchy object name to load from FEM_HIERARCHIES_T. Only those hierarchy objects of the designated dimension appear in the pick list.
- Hierarchy Definition
Pick a hierarchy definition to load from FEM_HIERARCHIES_T. Only those hierarchy definitions of the designated hierarchy object appear in the pick list.
- Execution Mode
Pick an execution mode of either *Snapshot* or *Error Reprocessing* from the pick list. If you pick *Snapshot*, the loader loads all records where the STATUS = 'LOAD'. If you pick *Error Reprocessing*, the loader loads only those records where the STATUS <> 'LOAD'.

Note: Status values must all be in upper case. If not, the loader programs may not pick up all the rows.

Output

When a load is successful, the loader removes all rows for the hierarchy definition from the various interface tables.

In addition to the standard Concurrent Manager error and log messages, the loader also updates the STATUS column for error rows in the source interface tables with a code indicating what was wrong with the record.

The following table lists the possible STATUS codes:

Status Code	Meaning
CHILD_WITH_MULTIPLE_PARENTS	The child has multiple parents for a hierarchy type that allows only a single parent.
CIRCULAR_REFERENCE	A child is its own parent through a relationship to other members.
GROUP_SEQ_RULE_VIOLATED	The parent/child relationship violated the Group Sequence order for the hierarchy.
INCONSISTENT_ROOT_NODE	The root node is consistently identified as such in the hierarchy. Some records have a 'Y', while others have an 'N'.
INVALID_CAL_PERIOD_ID	The CAL_PERIOD_ID that would be generated from the input data does not exist in Enterprise Performance Foundation.
INVALID_DIMENSION_LABEL	The Dimension Label does not exist in Enterprise Performance Foundation.
INVALID_GROUP_SEQ_ENFORCED_CODE	The Dimension Group Sequence Enforced Code is invalid.
INVALID_HIERARCHY	The hierarchy could not be loaded, thus preventing loading for the dependent data.
INVALID_LOWEST_LEVEL_MEMBER	The hierarchy has members at the lowest level that are not <i>Reconciliation Child</i> members for a hierarchy type that requires such.
INVALID_MEMBER	The dimension member does not exist in Enterprise Performance Foundation.
INVALID_ROOT_NODE	The identified root node is a child within the same hierarchy.
LOAD	The record is ready for loading.
MISSING_ROOT_NODE	The top-level root node cannot be replaced when reloading a hierarchy. The original root node must be used.
MULTIPLE_TOP	The hierarchy is designated as a single top hierarchy but has multiple tops in the parent/child relationships

Table Descriptions

FEM_HIERARCHIES_T

This table is the *header* interface table for hierarchies. This table identifies all of the hierarchy definitions possible for loading, as well as the properties of the associated hierarchy object.

Column Name	Datatype	Nullable	Column Description
HIERARCHY_OBJECT_NAME	VARCHAR2(150)	Not Null	Object name of the new hierarchy to be imported.
FOLDER_NAME	VARCHAR2(150)	Not Null	Designates the Folder for the new hierarchy.
LANGUAGE	VARCHAR2(4)	Not Null	Designates the Language for the Hierarchy Object Name and the Folder Name.
DIMENSION_VARCHAR_LABEL	VARCHAR2(30)	Not Null	Designates the Dimension of the Hierarchy.
HIERARCHY_TYPE_CODE	VARCHAR2(30)	Not Null	Designates the rules for defining parent/child relationships within the hierarchy. Possible types are: <ul style="list-style-type: none"> • OPEN • RECONCILIATION
GROUP_SEQUENCE_ENFORCED_CODE	VARCHAR2(30)	Not Null	Specifies whether a hierarchy uses Groups and is constrained by Group Sequence, or does not use Groups at all. The three choices are: <ul style="list-style-type: none"> • NO_GROUPS • SEQUENCE_ENFORCED • SEQUENCE_ENFORCED_SKIP_LEVEL
MULTI_TOP_FLAG	VARCHAR2(1)	Not Null	Designates if the topmost depth of the hierarchy is a single node, or if multiple nodes exist at the topmost depth.
MULTI_VALUE_SET_FLAG	VARCHAR2(1)	Not Null	Designates whether the hierarchy comprises members from multiple value sets. If 'Y', the hierarchy has multiple rows in FEM_HIER_VALUE_SETS. If 'N', then the hierarchy has a single row in FEM_HIER_VALUE_SETS.
HIERARCHY_USAGE_CODE	VARCHAR2(30)	Not Null	Designates how the hierarchy is to be used in the system. Choices are: <ul style="list-style-type: none"> • PLANNING (used only by Oracle Enterprise Planning and Budgeting) • STANDARD (available to all applications)

Column Name	Datatype	Nullable	Column Description
FLATTENED_ROWS_ FLAG	VARCHAR2(1)	Not Null	A 'Y' indicates that the hierarchical relationships are flattened (that is, exploded) in the HIER table. An 'N' indicates that only the parent/child rows are stored for the hierarchy.
STATUS	VARCHAR2(30)	Not Null	Designates the Status of the hierarchy row for loading. Records with a STATUS = 'LOAD' are loaded by the Dimension Hierarchy Loader. All other STATUS codes designate some sort of error from the load.
HIERARCHY_OBJ_ DEF_NAME	VARCHAR2(150)	Not Null	Identifies the Object Definition for the Hierarchy. If this is an existing Object Definition name, then the Dimension Hierarchy Loader replaces the parent/child records in the HIER table with the new parent/child relationship from the appropriate _T table.
EFFECTIVE_START_ DATE	DATE		Designates the date upon which the Hierarchy Object Definition becomes effective. Both the Start Date and End Date are validated by the Dimension Hierarchy Loader to verify that they do not conflict with any Start Date/End Date ranges of existing Hierarchy Object definitions.
EFFECTIVE_END_ DATE	DATE		Designates the date upon which the Hierarchy Object Definition is no longer effective. Both the Start Date and End Date are validated by the Dimension Hierarchy Loader to verify that they do not conflict with any Start Date/End Date ranges of existing Hierarchy Object definitions.

The primary key is on the following column:

- HIERARCHY_OBJECT_NAME

FEM_HIER_DIM_GRP_S_T

This interface table designates the levels (dimension groups) that apply to a new hierarchy. You must populate it when loading a new hierarchy from FEM_HIERARCHIES_T where the value for GROUP_SEQUENCE_ENFORCED_CODE is either SEQUENCE_ENFORCED or SEQUENCE_ENFORCED_SKIP_LEVEL.

The levels loaded in the group file determine the levels that are available for use in branches in the hierarchy. If you choose to load discontinuous levels, you can still load a hierarchy, and the sequence will be enforced for all loaded levels. For example, if a level sequence consists of levels A, B, C, and D, and you want to include only levels A, B, and D in the hierarchy, you must load all the levels that you need within that sequence (A, B, and D), and you can only include dimension members from levels A, B, and D in your hierarchy branches.

The SEQUENCE_ENFORCED option specifies that the loader is to verify that the dimension members in a hierarchy are linked to each other from the levels specified in the group loader table. The loader also verifies that members are linked in the absolute order of levels, as specified in dimension levels for that dimension. Thus, in the preceding example, A links to B and then to D, but D cannot link to A because - even though the valid levels are A, B and D - A is a higher level relative to B, C, and D.

In cases where SEQUENCE_ENFORCED_SKIP_LEVEL is specified, you must load all levels that are part of the sequence that you want to use, even if you are not going to use each level in each hierarchy branch. For example, suppose that a level sequence consists of levels A, B, C, and D, and you want to create hierarchy branches consisting of dimension members from levels A, C and D, where some branches link members from A to D directly. In this case, you must load levels A, C, and D in the group loader file. The skip level aspect is enforced among the levels, and the sequence enforced aspect is expressed in requiring members to link from A to C to D or from A to D.

Column Name	Datatype	Nullable	Column Description
DIMENSION_GROUP_DISPLAY_CODE	VARCHAR2(150)	Not Null	Identifies the Level (Dimension Group).
HIERARCHY_OBJECT_NAME	VARCHAR2(150)	Not Null	Object name of the new hierarchy to be imported.
LANGUAGE	VARCHAR2(4)	Not Null	Designates the Language of the DIMENSION_GROUP_NAME and HIERARCHY_OBJECT_NAME.
STATUS	VARCHAR2(30)	Not Null	Designates the Status of the row for loading. Records with a STATUS = 'LOAD' are loaded by the Dimension Hierarchy Loader. All other STATUS codes designate some sort of error from the load.

The primary key is on the following columns:

- DIMENSION_GROUP_DISPLAY_CODE
- HIERARCHY_OBJECT_NAME

FEM_HIER_VALUE_SETS_T

This interface table designates the Value Sets for a new hierarchy. It is populated only when loading a new hierarchy from FEM_HIERARCHIES_T.

Column Name	Datatype	Nullable	Column Description
HIERARCHY_OBJECT_NAME	VARCHAR2(150)	Not Null	Object name of the new hierarchy to be imported.
VALUE_SET_DISPLAY_CODE	VARCHAR2(150)	Not Null	Alphanumeric code of the Value Set.
LANGUAGE	VARCHAR2(4)	Not Null	Designates the Language of the HIERARCHY_OBJECT_NAME.
STATUS	VARCHAR2(30)	Not Null	Designates the Status of the row for loading. Records with a STATUS = 'LOAD' are loaded by the Enterprise Performance Foundation Dimension Hierarchy Loader. All other STATUS codes designate some sort of error from the load.

The primary key is on the following columns:

- HIERARCHY_OBJECT_NAME
- VALUE_SET_DISPLAY_CODE

FEM_xName_HIER_T (Value Set Dimensions parent/child relationship table)

This interface table stores the parent/child relationships for hierarchy definitions. The parent/child relationships are identified by a hierarchy object name and hierarchy definition name. The Dimension Hierarchy Loader uses the combination of hierarchy object name and hierarchy definition name for the designated LANGUAGE to identify the particular Hierarchy Object Definition for the parent/child rows. If the loader finds an existing hierarchy definition, it completely replaces it with the new parent/child rows in the interface table. Otherwise, the loader creates a new hierarchy definition for the hierarchy object name.

The HIER_T table structure for the Calendar Period dimension is different than that of the Value Set dimensions. Refer to the FEM_CAL_PERIODS_HIER_T section for details on the parent/child relationship table for the Calendar Period dimension.

Note: Only the *connect by* relationship rows are stored in the _HIER_T table, along with parent=child rows to identify any root nodes. The Dimension Hierarchy Loader creates the additional *flattened/exploded* rows in the FEM_xName_HIER table only for hierarchies with the FLATTENED_ROWS_FLAG = 'Y'.

Note: Oracle Applications does not require flattening for any hierarchies at this time. This feature is available for performance reasons for future applications using Enterprise Performance Foundation.

Example names for these interface tables are as follows:

- FEM_PRODUCTS_HIER_T
- FEM_CHANNELS_HIER_T
- FEM_NAT_ACCTS_HIER_T

Column Name	Datatype	Nullable	Column Description
HIERARCHY_ OBJECT_NAME	VARCHAR2(150)	Not Null	Object name of the hierarchy to be imported.
HIERARCHY_OBJ_ DEF_DISPLAY_ NAME	VARCHAR2(150)	Not Null	Display name of the object definition of the hierarchy to be imported.
PARENT_DISPLAY_ CODE	VARCHAR2(150)	Not Null	Identifies the dimension member that is the parent.
PARENT_VALUE_ SET_DISPLAY_ CODE	VARCHAR2(150)	Not Null	Alphanumeric code for the Value Set of the parent member.
CHILD_DISPLAY_ CODE	VARCHAR2(150)	Not Null	Identifies the child dimension member.
CHILD_VALUE_ SET_DISPLAY_ CODE	VARCHAR2(150)	Not Null	Alphanumeric code for the Value Set of the child value set.
LANGUAGE	VARCHAR2(4)	Not Null	Designates the language code for the HIERARCHY_ OBJECT_NAME and HIERARCHY_ OBJ_DEF_DISPLAY_ NAME columns.
DISPLAY_ORDER_ NUM	NUMBER	Not Null	Designates the order in which the relationships appear in the UI. The display order is relative to the Parent.

Column Name	Datatype	Nullable	Column Description
WEIGHTING_PCT	NUMBER(3,2)		Designates a weighting percentage for the relationship.
STATUS	VARCHAR2(30)	Not Null	Designates the Status of the row for loading. Records with a STATUS = 'LOAD' are loaded by the Enterprise Performance Foundation Dimension Hierarchy Loader. All other STATUS codes designate some sort of error from the load. Note that the <i>Reprocess Errors Flag</i> in FEM_DIM_LOAD_STATUS enables users to load records that have an error status without requiring them to update the STATUS column back to 'LOAD'.

The primary key is on the following columns:

- HIERARCHY_OBJECT_NAME
- HIERARCHY_OBJ_DEF_DISPLAY_NAME
- PARENT_DISPLAY_CODE
- PARENT_VALUE_SET_ID
- CHILD_DISPLAY_CODE
- CHILD_VALUE_SET_ID

FEM_CAL_PERIODS_HIER_T (Calendar Period parent/child relationship table)

This interface table stores the parent/child relationships for Calendar Period dimension hierarchies. The parent/child relationships are identified by a hierarchy object name and hierarchy definition name. The Dimension Hierarchy Loader uses the combination of hierarchy object name and definition name for the designated LANGUAGE to identify the particular Hierarchy Object Definition for the parent/child rows. If the loader finds an existing hierarchy definition, it completely replaces it with the new parent/child rows in the interface table. Otherwise, the loader creates a new hierarchy definition for the hierarchy object name.

Note: Only the *connect by* relationship rows are stored in the _HIER_T table, along with parent=child rows to identify any root nodes. The Dimension Hierarchy Loader creates the additional *flattened/exploded* rows in the FEM_CAL_PERIODS_HIER table only for hierarchies with the FLATTENED_ROWS_FLAG = 'Y'.

Column Name	Datatype	Nullable	Column Description
HIERARCHY_OBJECT_NAME	VARCHAR2(150)	Not Null	Object name of the hierarchy to be imported.
HIERARCHY_OBJ_DEF_DISPLAY_NAME	VARCHAR2(150)	Not Null	Display name of the object definition of the hierarchy to be imported.
CALENDAR_DISPLAY_CODE	VARCHAR2(150)	Not Null	It identifies the Calendar for both parent and child nodes.
PARENT_DIM_GRP_DISPLAY_CODE	VARCHAR2(150)	Not Null	Identifies the Level (Dimension Group) for the parent member.
PARENT_END_DATE	DATE	Not Null	Ending date for the parent period.
PARENT_PERIOD_NUMBER	NUMBER(15)	Not Null	This is the Period Number for the parent period. For example, for MONTH, January = 1, February = 2, March = 3, and so on. For Days, Jan-01 = 1, Jan-02 = 2, Jan-03=3, and so on. This number resets for each year.
CHILD_DIM_GRP_DISPLAY_CODE	VARCHAR2(150)	Not Null	Identifies the Level (Dimension Group) for the child member.
CHILD_END_DATE	DATE	Not Null	Ending date for the child period.
CHILD_PERIOD_NUMBER	NUMBER(15)	Not Null	This is the Period Number for the child period. For example, for MONTH, January = 1, February = 2, March = 3, and so on. For Days, Jan-01 = 1, Jan-02 = 2, Jan-03=3, and so on. This number resets for each year.
LANGUAGE	VARCHAR2(4)	Not Null	Designates the language code for the HIERARCHY_OBJECT_NAME and HIERARCHY_OBJ_DEF_DISPLAY_NAME columns.

Column Name	Datatype	Nullable	Column Description
DISPLAY_ORDER_NUM	NUMBER	Not Null	Designates the order in which the relationships appear in the UI. The display order is relative to the Parent.
WEIGHTING_PCT	NUMBER(3,2)		Designates a weighting percentage for the relationship.
STATUS	VARCHAR2(30)	Not Null	Designates the Status of the row for loading. Records with a STATUS = 'LOAD' are loaded by the Enterprise Performance Foundation Dimension Hierarchy Loader. All other STATUS codes designate some sort of error from the load. Note that the <i>Reprocess Errors Flag</i> in FEM_DIM_LOAD_STATUS enables users to load records that have an error status without requiring them to update the STATUS column back to 'LOAD'.

The primary key is on the following columns:

- HIERARCHY_OBJECT_NAME
- HIERARCHY_OBJ_DEF_DISPLAY_NAME
- CALENDAR_DISPLAY_CODE
- PARENT_DIM_GRP_DISPLAY_CODE
- PARENT_END_DATE
- PARENT_PERIOD_NUMBER
- CHILD_DIM_GRP_DISPLAY_CODE
- CHILD_END_DATE
- CHILD_PERIOD_NUMBER

Sample SQL*Loader DAT and CTL Files for the Hierarchy Interface Tables

The following section provides examples of SQL*Loader DAT and CTL files for loading into the Hierarchy interface tables.

Hierarchy Properties Interface Table

- **CTL file for FEM_HIERARCHIES_T**

```
OPTIONS (ERRORS=0, SILENT=(FEEDBACK), DIRECT=TRUE)
LOAD DATA
INFILE 'hierarchies_t.dat'
BADFILE 'hierarchies_t.bad'
DISCARDFILE 'hierarchies_t.dsc'
INTO TABLE FEM_HIERARCHIES_T
REPLACE
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
(
HIERARCHY_OBJECT_NAME,
FOLDER_NAME,
LANGUAGE,
DIMENSION_VARCHAR_LABEL,
HIERARCHY_TYPE_CODE,
GROUP_SEQUENCE_ENFORCED_CODE,
MULTI_TOP_FLAG,
MULTI_VALUE_SET_FLAG,
HIERARCHY_USAGE_CODE,
FLATTENED_ROWS_FLAG,
HIER_OBJ_DEF_DISPLAY_NAME,
STATUS,
CALENDAR_DISPLAY_CODE
)
```

- **DAT file for FEM_HIERARCHIES_T**

```
"FMS.TIME","Data Integration","US","CAL_PERIOD","OPEN",
"SEQUENCE_ENFORCED","Y","N","STANDARD","N",
"Application Time Hierarchy","LOAD","Standard","HI.EPB.LINE",
"Data Integration","US","LINE_ITEM","RECONCILIATION",
"SEQUENCE_ENFORCED","Y","N","STANDARD","N",
"Line Item Standard Hierarchy","LOAD",""
```

Hierarchy Levels Interface Table

- **CTL file for FEM_HIER_DIM_GRPST**

```
OPTIONS (ERRORS=0, SILENT=(FEEDBACK), DIRECT=TRUE)
LOAD DATA
INFILE 'hier_dim_grps_t.dat'
BADFILE 'hier_dim_grps_t.bad'
DISCARDFILE 'hier_dim_grps_t.dsc'
INTO TABLE FEM_HIER_DIM_GRPST
REPLACE
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
(
HIERARCHY_OBJECT_NAME,
LANGUAGE,
DIMENSION_GROUP_DISPLAY_CODE,
STATUS
)
```

- **DAT file for FEM_HIER_DIM_GRPST**

```
"FMS.TIME", "US", "CAL_PERIOD_L1", "LOAD"  
"FMS.TIME", "US", "CAL_PERIOD_L2", "LOAD"
```

Hierarchy Value Sets Interface Table

- **CTL file for FEM_HIER_VALUE_SETS_T**

```
OPTIONS (ERRORS=0, SILENT=(FEEDBACK), DIRECT=TRUE)  
LOAD DATA  
INFILE 'hier_value_sets_t.dat'  
BADFILE 'hier_value_sets_t.bad'  
DISCARDFILE 'hier_value_sets_t.dsc'  
INTO TABLE FEM_HIER_VALUE_SETS_T  
REPLACE  
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '''  
(  
HIERARCHY_OBJECT_NAME,  
VALUE_SET_DISPLAY_CODE,  
LANGUAGE,  
STATUS  
)
```

- **DAT file for FEM_HIER_VALUE_SETS_T**

```
"HI.EPB.LINE", "LINE_ITEM", "US", "LOAD"  
"HI.AA75213", "USER_DIM5", "US", "LOAD"
```

Line Item Hierarchy Relationship Interface Table

- **CTL file for FEM_LN_ITEMS_HIER_T**

```
OPTIONS (ERRORS=0, SILENT=(FEEDBACK), DIRECT=TRUE)  
LOAD DATA  
INFILE 'line_item_hier_t.dat'  
BADFILE 'line_item_hier_t.bad'  
DISCARDFILE 'line_item_hier_t.dsc'  
INTO TABLE FEM_LN_ITEMS_HIER_T  
REPLACE  
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '''  
(  
HIERARCHY_OBJECT_NAME,  
HIERARCHY_OBJ_DEF_DISPLAY_NAME,  
PARENT_DISPLAY_CODE,  
PARENT_VALUE_SET_DISPLAY_CODE,  
CHILD_DISPLAY_CODE,  
CHILD_VALUE_SET_DISPLAY_CODE,  
DISPLAY_ORDER_NUM,  
LANGUAGE,  
STATUS  
)
```

- **DAT file for FEM_LN_ITEMS_HIER_T**

```

"HI.EPB.LINE", "Line Item Standard Hierarchy", "LN0001",
"LINE_ITEM", "LN0001", "LINE_ITEM", "100", "US", "LOAD"
"HI.EPB.LINE", "Line Item Standard Hierarchy", "LN0001",
"LINE_ITEM", "LN0002", "LINE_ITEM", "1", "US", "LOAD"

```

Calendar Period Hierarchy Relationship Interface Table

- **CTL file for FEM_CAL_PERIODS_HIER_T**

```

OPTIONS (ERRORS=0, SILENT=(FEEDBACK), DIRECT=TRUE)
LOAD DATA
INFILE 'cal_period_hier_t.dat'
BADFILE 'cal_period_hier_t.bad'
DISCARDFILE 'cal_period_hier_t.dsc'
INTO TABLE FEM_CAL_PERIODS_HIER_T
REPLACE
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
(
HIERARCHY_OBJECT_NAME,
HIERARCHY_OBJ_DEF_DISPLAY_NAME,
CALENDAR_DISPLAY_CODE,
PARENT_CAL_PERIOD_END_DATE,
PARENT_CAL_PERIOD_NUMBER,
PARENT_DIM_GRP_DISPLAY_CODE,
CHILD_CAL_PERIOD_END_DATE,
CHILD_CAL_PERIOD_NUMBER,
CHILD_DIM_GRP_DISPLAY_CODE,
DISPLAY_ORDER_NUM,
LANGUAGE,
STATUS
)

```

- **DAT file for FEM_CAL_PERIODS_HIER_T**

```

"FMS.TIME", "Application Time Hierarchy", "Standard", "31-DEC-2004",
"1", "CAL_PERIOD_L1", "31-DEC-2004", "1",
"CAL_PERIOD_L1", "100", "US", "LOAD"
"FMS.TIME", "Application Time Hierarchy", "Standard", "31-DEC-2004",
"1", "CAL_PERIOD_L1", "31-MAR-2004", "1",
"CAL_PERIOD_L2", "1", "US", "LOAD"

```

External General Ledger Data Loader

The External General Ledger Data Loader program is part of the Enterprise Performance Foundation integration process that is responsible for posting balance data from the interface table to FEM_BALANCES. After you have loaded data into the interface table and submitted the posting process, all operations occur within the database and require no further user interaction.

Note: You can load only one set of data into a specific ledger and calendar period for the FEM_BALANCES table. For example, if you load data into Ledger1, Dataset1, and calendar period JAN-05, you cannot load additional data for Ledger1 and JAN-05 into a different data set in the FEM_BALANCES table. The FEM_BALANCES table is secured so that only a single source of truth for a ledger and calendar period is

provided. To load data for a given combination of ledger and calendar period into multiple data sets, use the FEM_DATAx (FEM_DATA1 – FEM_DATA20) tables, as they do not have the same restriction as the FEM_BALANCES table.

System Flow

You must perform the following steps during the system flow:

1. Load balance data into the interface table.
2. Load all dimension values and attributes used in the balance data into the dimension interface tables.
3. Run the Dimension Member Loader process to load the dimension values into Enterprise Performance Foundation.
4. Run the External General Ledger Data Loader program to load balances from the interface table to FEM_BALANCES.

You can repeat this process as often as your business process requires. You need to perform steps 2 and 3 only if you are using new dimension values.

External General Ledger Interface Table Descriptions

This section describes the kinds of external general ledger interface tables.

FEM_BAL_INTERFACE_T

FEM_BAL_INTERFACE_T is the table you use to import balances into FEM_BALANCES through the Enterprise Performance Foundation integration process. You must supply values for all NOT NULL columns.

Before loading for the first time, you need to create the unique processing key index on FEM_BALANCES and its interface table (FEM_BAL_INTERFACE_T), and you need to define the processing key on FEM_BALANCES. However, you do not need to define the processing key for the FEM_BAL_INTERFACE_T table.

The unique index on FEM_BAL_INTERFACE_T should always contain the following columns (in order):

1. CAL_PERIOD_NUMBER
2. CAL_PERIOD_END_DATE
3. CAL_PER_DIM_GRP_DISPLAY_CODE
4. LEDGER_DISPLAY_CODE
5. DS_BALANCE_TYPE_CODE
6. LOAD_SET_ID
7. CCTR_ORG_DISPLAY_CODE
8. Other Display Code columns corresponding to FEM_BALANCES processing key columns
9. SOURCE_SYSTEM_DISPLAY_CODE
10. CURRENCY_TYPE_CODE

- 11. CURRENCY_CODE
- 12. BUDGET_DISPLAY_CODE
- 13. ENCUMBRANCE_TYPE_CODE
- 14. FINANCIAL_ELEM_DISPLAY_CODE

The following table contains columns for Column Name, Data Type, Size, Not Null, and Column Description:

Column Name	Data Type	Size	Not Null	Column Description
LOAD_SET_ID	NUMBER		Not Null	User-defined Data Group identifying a particular load
LOAD_METHOD_CODE	VARCHAR2	30	Not Null	Operation mode for the record. This is used for the External General Ledger Data Loader program for processing purposes. Possible values: <ul style="list-style-type: none"> • S - Snapshot • I - Incremental
BAL_POST_TY PE_CODE	VARCHAR2	30	Not Null	Indicates how balance should be posted Possible values: <ul style="list-style-type: none"> • R – Replace Existing Balance • A – Add to Existing Balance Use value A for all snapshot rows (LOAD_METHOD_CODE value S) to prevent potential errors in loading order.
CAL_PER_DIM_GRP_DISPLAY_CODE	VARCHAR2	150	Not Null	Calendar Period Dimension Group

Column Name	Data Type	Size	Not Null	Column Description
CAL_PERIOD_NUMBER	NUMBER	15	Not Null	Calendar Period Number provides alternate identification for the Calendar Period, together with Ledger ID, Calendar Period End Date, and Calendar Period Dimension Group Name.
CAL_PERIOD_END_DATE	DATE		Not Null	The end date of the calendar period
CCTR_ORG_DISPLAY_CODE	VARSHAR2	150	Not Null	Unique display code for the Company Cost Center Organization dimension value
CURRENCY_CODE	VARCHAR2	15	Not Null	Currency
CURRENCY_TYPE_CODE	VARCHAR2	30	Not Null	Currency Type Possible values: <ul style="list-style-type: none"> ENTERED TRANSLATED
DS_BALANCE_TYPE_CODE	VARCHAR2	30	Not Null	Balance Type Possible values: <ul style="list-style-type: none"> ACTUAL BUDGET ENCUMBRANCE
SOURCE_SYSTEM_DISPLAY_CODE	VARCHAR2	150	Not Null	Unique display code for the Source System dimension value
LEDGER_DISPLAY_CODE	VARCHAR2	150	Not Null	Unique display code for the Ledger dimension value

Column Name	Data Type	Size	Not Null	Column Description
BUDGET_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the Budget dimension value. Required if DS_BALANCE_TYPE_CODE is B.
ENCUMBRANCE_TYPE_CODE	VARCHAR2	150	Null	Unique display code for the Encumbrance dimension value. Required if DS_BALANCE_TYPE_CODE is E.
FINANCIAL_ELEMENT_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the Financial Element dimension value
PRODUCT_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the Product dimension value
NATURAL_ACCOUNT_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the Natural Account dimension value
CHANNEL_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the Channel dimension value
LINE_ITEM_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the Line Item dimension value
PROJECT_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the Project dimension value
CUSTOMER_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the Customer dimension value
ENTITY_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the Entity dimension value
INTERCOMPANY_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the Intercompany dimension value

Column Name	Data Type	Size	Not Null	Column Description
TASK_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the Task dimension value
USER_DIM1_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the user-defined dimension value
USER_DIM2_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the user-defined dimension value
USER_DIM3_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the user-defined dimension value
USER_DIM4_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the user-defined dimension value
USER_DIM5_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the user-defined dimension value
USER_DIM6_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the user-defined dimension value
USER_DIM7_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the user-defined dimension value
USER_DIM8_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the user-defined dimension value
USER_DIM9_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the user-defined dimension value
USER_DIM10_DISPLAY_CODE	VARCHAR2	150	Null	Unique display code for the user-defined dimension value
XTD_BALANCE_END	NUMBER		Null	Period-to-Date end or average balance in the entered currency

Column Name	Data Type	Size	Not Null	Column Description
XTD_BALANCE_ F	NUMBER		Null	Period-to-Date end or average balance in the functional currency for the current set of books
YTD_BALANCE_ E	NUMBER		Null	Year-to-Date end or average balance as of end date for the period in the entered currency
YTD_BALANCE_ F	NUMBER		Null	Year-to-Date end or average balance as of end date for the period in the functional currency for the current set of books
QTD_ BALANCE_E	NUMBER		Null	Quarter-to-Date end or average balance as of end date for the period in the entered currency
QTD_ BALANCE_F	NUMBER		Null	Quarter-to-Date end or average balance as of end date for the period in the entered currency for the current set of books
PTD_DEBIT_ BALANCE_E	NUMBER		Null	Debit Amount in the entered currency.
PTD_CREDIT_ BALANCE_E	NUMBER		Null	Credit Amount in the entered currency.
YTD_DEBIT_ BALANCE_E	NUMBER		Null	Year-to-Date debit amount in the entered currency.
YTD_CREDIT_ BALANCE_E	NUMBER		Null	Year-to-Date credit amount in the entered currency.

Column Name	Data Type	Size	Not Null	Column Description
POSTING_REQUEST_ID	NUMBER		Null	For Internal Use only. This is the Concurrent Request ID of the loading process that is currently loading this record.
POSTING_ERROR_CODE	VARCHAR2	30	Null	For Internal Use only. This indicates the error status of the record. This column is NULL for no error or not processed yet. Not null values indicate why the row could not be loaded.
PREVIOUS_ERROR_FLAG	VARCHAR2	1	Null	For Internal Use only. This indicates whether the record has an outstanding error at the start of the posting execution.

Populating the Interface Tables for the External General Ledger Data Loader

The following section provides information on how to populate the External General Ledger Data Loader interface tables in preparation for a load.

Specifying Time

Time is specified in the interface record using the following columns:

- CAL_PER_DIM_GRP_DISPLAY_CODE
This is the Dimension Group in which the Calendar Period belongs.
- CAL_PERIOD_END_DATE
This is the end date of the Calendar Period.
- CAL_PERIOD_NUMBER
This is the GL period number of the Calendar Period.

The following table contains columns for Calendar Period End Date, Level (Dimension Group Display Code), and Calendar Period Number:

CAL_PERIOD_END_DATE	CAL_PER_DIM_GRP_DISPLAY_CODE	CAL_PERIOD_NUMBER
31-May-2004	Month	12
30-Jun-2004	Month	1
31-Jul-2004	Month	2
31-Aug-2004	Month	3
30-Sep-2004	Month	4
31-Oct-2004	Month	5
30-Nov-2004	Month	6
31-Dec-2004	Month	7
31-Jan-2005	Month	8
28-Feb-2005	Month	9
31-Mar-2005	Month	10
30-Apr-2005	Month	11
31-May-2005	Month	12
31-Aug-2004	Quarter	1
30-Nov-2004	Quarter	2
28-Feb-2005	Quarter	3
31-May-2005	Quarter	4
31-May-2005	Year	1
30-Jun-2005	Month	1

Dealing with Entered and Translated Balances

The External General Ledger Data Loader program processes both entered balances and translated balances present in the interface table. The column `CURRENCY_TYPE_CODE` is used to distinguish between the two types. You should use `CURRENCY_TYPE_CODE` of *ENTERED* for entered or cumulative functional balances and *TRANSLATED* for translated balances.

The balance information required for these two types of balances is as follows:

- Entered Balances for a given currency must be specified in the interface record as follows:
 - `CURRENCY_CODE` is the currency used in entering the original transactions.
 - `CURRENCY_TYPE_CODE` should be *ENTERED*.

The functional equivalent of the entered balances must be present. This means that the BALANCE_F columns must be filled in.

Oracle recommends that the BALANCE_E column be filled in because it contains the balances entered in the given currency.

- Cumulative Functional Balances must be specified in the interface record as follows:
 - CURRENCY_CODE is the functional currency.
 - CURRENCY_TYPE_CODE should be *ENTERED*.
 - The cumulative functional balance must be entered in the BALANCE_F columns.
 - The BALANCE_E columns may be left NULL, or the cumulative functional value may be repeated here.
- Translated Balances for a given currency must be specified in the interface record as follows:
 - CURRENCY_CODE is balances translated to currency.
 - CURRENCY_TYPE_CODE should be *TRANSLATED*.

Resulting translated balances must be provided in the BALANCE_E columns.

Any non-required balance columns populated in the interface record are loaded into FEM_BALANCES without further validation.

Period Specific QTD and YTD Data

You can provide the running QTD and YTD balances for each period by populating the QTD_BALANCE_E, QTD_BALANCE_F, YTD_BALANCE_E, and YTD_BALANCE_F columns for each interface record. The External General Ledger Data Loader program checks for data in these columns based on the user input parameter, *Period Specific Amounts Provided*.

Parameter Value	Columns Expected to Be Populated
Populate PTD only	XTD_BALANCE_E, XTD_BALANCE_F
Populate PTD and QTD only	XTD_BALANCE_E, XTD_BALANCE_F, QTD_BALANCE_E, QTD_BALANCE_F
Populate PTD and YTD only	XTD_BALANCE_E, XTD_BALANCE_F, TD_BALANCE_E, YTD_BALANCE_F
Populate PTD, QTD, and YTD	XTD_BALANCE_E, XTD_BALANCE_F, QTD_BALANCE_E, QTD_BALANCE_F, TD_BALANCE_E, YTD_BALANCE_F

The program validates each balance pair based on the entered balance/translated balance rules listed in the previous section.

Account Type and Signage

The External General Ledger Data Loader program does not support signage conversion based on account types. You must populate your data with the right signage.

Data Load Execution Modes

The External General Ledger Data Loader program supports two execution modes, Snapshot and Incremental. You specify the execution mode when you submit the data load run. At the data level, you distinguish data for Snapshot loads from data for Incremental loads using the column `LOAD_METHOD_CODE`.

The External General Ledger Data Loader program performs a straight INSERT into `FEM_BALANCES` when running in Snapshot mode for performance benefits. This mode assumes that no data exists in `FEM_BALANCES` for the given Ledger, Period, and Dataset combination. The process registration routine checks that a snapshot can be run only when there are no prior successfully completed runs recorded in the system for the given Ledger, Period, and Database.

A snapshot load may terminate in error rerun status if, for example, all data rows failed validation or there was a database error. In this case, the snapshot load can be rerun, in snapshot mode, after the data errors have been corrected or the cause of the database error has been resolved.

When the program is running in Incremental mode, it merges new data into the existing data in `FEM_BALANCES`. You can choose to replace the existing data or add to the existing data at the row level by populating the `BAL_POST_TYPE_CODE` column for each interface record. A value 'R' represents a replace, while a value 'A' represents an addition.

Processing Requirements

The External General Ledger Data Loader program observes the following processing requirements from the Enterprise Performance Foundation architecture:

- For each Ledger, Period, and Balance type (Actuals, individual budgets, or individual Encumbrance types) combination, data can be posted into only one dataset.
- For each Ledger and Period combination, balances must be processed by only one version of the same XGL integration rule.
- Two loads for the same Ledger, Dataset, and Period cannot run at the same time.
- You must start data load for a period with a Snapshot load first. No Incremental loads are allowed until a Snapshot load has been run.
- For each period, only one Snapshot load is allowed, and it has to be the first load.
- Snapshot loads can be run only for the latest loaded period. Periods can be skipped, but snapshot loads cannot be run out of sequence.
- The Dataset runtime parameter for the External General Ledger Data Loader program is also used for selecting rows for loading. Each data set has a value of 'ACTUAL', 'BUDGET', OR 'ENCUMBRANCE' for the `DATASET_BALANCE_TYPE_CODE` attribute. Note the following related considerations:
 - For Actual data sets, rows for which the value for `DATASET_BALANCE_TYPE_CODE` is 'ACTUAL' are selected from the interface table.
 - For Budget data sets, the Budget runtime parameter must also be supplied, and rows are selected from the interface table where the value for `DATASET_BALANCE_TYPE_CODE` is 'BUDGET' and the value for `BUDGET_DISPLAY_CODE` is the display code for the Budget parameter.

- For Encumbrance data sets, the Encumbrance Type runtime parameter must also be supplied, and rows are selected from the interface table where the value for DATASET_BALANCE_TYPE_CODE is 'ENCUMBRANCE' and the value for ENCUMBRANCE_TYPE_CODE is the display code for the Encumbrance parameter.

Data Load by Load Sets

You can load multiple sets of incremental balance data before submitting a data load run. This creates an interesting situation where the order in which the data is posted actually matters.

For instance, you first populated the interface table with an addition of \$10 for dimension combination A and then decided to replace the entire balance of A with \$500. If you first delete the \$10 addition row from the interface table before loading the second row, everything would work just fine. You can, however, leave both rows in the interface table. Thus, it is important that the program can determine that it should first post the \$10 addition, before replacing the whole balance with \$500.

Therefore, some indication of the post order is necessary at the data level, and so you have the concept of a Load Set, which is represented by the LOAD_SET_ID column. Each load set contains the group of balance rows that need to be processed together.

For Incremental loads, the External General Ledger Data Loader program scans the list of load sets in the interface table and starts processing them in ascending order of the LOAD_SET_ID column.

You should use a unique value for LOAD_SET_ID for each batch of balances data loaded into the interface table.

Parallel Processing

Data load requests submitted for the same Ledger, Period, and Dataset combinations cannot be run in parallel. This is to avoid multiple requests trying to process the same set of data.

Data load requests for different Ledgers, Periods, or Datasets can be run in parallel, however. This is because they are processing different sets of interface rows and writing to different sets of rows in FEM_BALANCES.

This processing mechanism is enforced first by the conflict domain at the Concurrent Manager level, as well as the process registration logic within the External General Ledger Data Loader program.

Moreover, data in the interface table is marked with the Concurrent Request ID of the data load process so you can load new rows into the interface table while load processes are running.

Loading Processing Key Dimension Information

All FEM_BALANCES processing key dimensions have to be populated in the interface table.

Error Reprocessing

The External General Ledger Data Loader program reprocesses all error records that you have fixed. Based on the execution mode of the data load request, the program not only processes new data but also picks up records for error reprocessing.

The following table contains columns for Execution Mode, LOAD_METHOD_CODE Filter, and Additional Row Conditions. It illustrates how the load program determines the set of interface records to load:

Execution Mode	LOAD_METHOD_CODE Filter	Additional Row Conditions
Snapshot	S	None. The program picks up both new and existing error snapshot rows. Error rows exist only if this is a rerun of a failed execution.
Incremental	I and S	For LOAD_METHOD_CODE of I, the program picks up both new and error rows. For LOAD_METHOD_CODE of S, the program picks up only error rows.

As mentioned in the previous sections, a Snapshot load can be submitted only once for each Ledger, Period, and Dataset combination and it has to be the first load. For a Snapshot load, there should never be any error records to reprocess. This is true in most cases. The only exception is when the Snapshot load fails in error rerun status, as described in the *Data Load Execution Modes* section. In this case, you are allowed to re-submit the Snapshot load, which picks up all snapshot records (new or previously failed) for processing.

For Incremental loads, the program picks both new incremental records and any error records (snapshot or incremental) for data load. The program reprocesses the snapshot error records, before moving onto the incremental records, and loads them in ascending order of their LOAD_SET_ID.

Error Tracking and Monitoring

Currently, any data error is tracked in the POSTING_ERROR_CODE column in each record. When a data error is detected, the program updates this column with the appropriate error code. The program request completes in a warning status with a message saying that data errors are detected. Only the first data error for each row is reported.

Failure Versus Warning Completion Status

As already mentioned, the data load request can complete with Failure or Warning status.

The request completes with a Failure status when any one of the following events happens:

- Unexpected database error
- Any process registration violation, such as attempting to run an Incremental load before a Snapshot load is run
- All records for a Snapshot load have failed data validation, causing nothing to be loaded into FEM_BALANCES
- The number of records in FEM_BALANCES affected by the data load request does not equal the number of records processed in the interface table. This indicates a corruption in FEM_BALANCES.

The request completes with a Warning status when any invalid data is found during processing.

Removing Loaded Data

- If you need to reload a Snapshot, you need to remove all of the existing data in FEM_BALANCES, including subsequent Incremental loads, for the given Ledger, Period, and Dataset. Incremental loads can be removed only by removing all of the data for the given Ledger, Dataset, and Period, including the Snapshot load.
- When you try to remove a data load in FEM_BALANCES, any records in the interface table associated with that data load should be deleted as well.

The Executed Rule Removal program handles this situation and deletes records from both FEM_BALANCES and FEM_BAL_INTERFACE_T.

Data Validation Errors

When a row in the interface table contains insufficient or invalid data, the External General Ledger Data Loader program populates the Posting Error Code column for that row with one of these values.

Data Load Error Code Values	Meaning
FEM_GL_POST_DUP_PROC_KEYS FEM_BAL_INTERFACE_T	Error: Duplicate rows are found in a unique combination of the processing key for dimensions.
FEM_GL_POST_INVALID_CURR_TYPE	Error: The Currency Type Code is not valid. Use valid Currency Type Code (ENTERED or TRANSLATED).
FEM_GL_POST_INVALID_DIM_MEMBER	Error: Invalid or missing dimension members are found in this record.
FEM_GL_POST_INVALID_POST_TYPE	Error: The Balance Post Type Code is not valid. Use a valid Balance Post Type Code: 'A' (Add) or 'R' (Replace).
FEM_GL_POST_MISSING_BAL_COL	Error: According to the QTD-YTD parameter and the currency type specification, some values for the XTD/QTD/YTD columns are missing.
FEM_GL_POST_MIX_LOAD_METHOD	Error: There is a mix of load methods in the current load set.
FEM_GL_POST_PREV_SET_ERROR	Error: This processing key combination has an error in a previous load set. That error must be corrected before its balance can be loaded.
FEM_GL_POST_TO_BE_REPROCESSED	Error: An unexpected database error has occurred. This record is processed again during the next execution.

Error rows are not deleted from the interface table but are left there for reprocessing by future runs after the errors have been corrected. Correct the errors in each row by supplying the missing values, by correcting the invalid values, or by loading the missing dimension values using the Dimension Loader program. Alternatively, you can delete the error rows from the interface table and reload them as corrected rows.

Program Parameters

This program is used for loading monetary and statistical data from external general ledger systems into the FEM_BALANCES table. The External General Ledger Data Loader program accepts the following parameters:

Parameter Name	Required	Description	Validations
Execution Mode	Y	Loading Execution Mode (Snapshot or Incremental)	
Ledger	Y	Ledger to load into	This value must exist in the Ledger dimension.
Calendar Period	Y	Calendar period to load into	This value must exist in the Calendar Period dimension, and its Calendar ID must match the Calendar ID assigned to the Ledger.
Budget	N	Budget to load into	This value must exist in the Budget dimension and be assigned as the Budget ID attribute for the Dataset. The value assigned to the Budget Ledger Attribute value must match the Ledger parameter.
Encumbrance Type	N	Encumbrance type to load for	This value must exist in the Encumbrance Type dimension and be assigned to the Encumbrance Type ID attribute of the Dataset.

Parameter Name	Required	Description	Validations
Dataset	Y	Dataset to load into	This value must exist in the dataset dimension. When the Balance Type Code attribute of the Dataset is <i>ACTUALS</i> , neither the Budget nor the Encumbrance Type parameters should be entered. When it is <i>BUDGET</i> , only the Budget parameter should be entered. When it is <i>ENCUMBRANCE</i> , only the Encumbrance type parameter should be entered.
Period Specific Amounts Provided	Y	Types of period-specific amounts provided: <ul style="list-style-type: none"> • <i>PTD Only</i> indicates that neither QTD nor YTD columns should be populated in FEM_BALANCES. • <i>PTD and YTD</i> indicates that only PTD and YTD columns should be populated. • <i>PTD and QTD</i> indicates that only PTD and YTD columns should be populated. • <i>PTD, QTD, and YTD</i> indicates that PTD, QTD, and YTD should be populated. 	

Note: For parameters based on a dimension's List of Values (LOV), only *enabled, global* (not *personal*) values show in the parameter's LOV.

Preliminary Steps for the Load Process

Before you begin the load process, complete the following steps:

1. Create a unique index on FEM_BALANCES.
2. Register the processing key for FEM_BALANCES using the newly-created unique index.
3. Create a unique index on FEM_BAL_INTERFACE_T that includes all the columns listed for it in the *External General Ledger Interface Table* section, including all the Display Code columns that correspond to a FEM_BALANCES processing key column, and set these columns to NOT NULL in the interface table.
4. Use Dimension and Hierarchy Management to define Ledgers, Periods, and Datasets, along with all of their required attributes, as described in the *Program Parameters* section. Alternatively, you can load these dimensions and their attributes using the Dimension Member Loader.

Loading Data into FEM_BALANCES

1. Load balances data into the FEM_BAL_INTERFACE_T table as described in the *External General Ledger Interface Table* and the *Populating the Interface Table* sections of this document.
2. For all other dimension values used for the balances data in the interface table, use Dimension and Hierarchy Management or the Dimension Member Loader to create or load these dimension members in their respective dimension member tables within the Enterprise Performance Foundation database. Once these values are created or loaded, you need to perform this step only for new dimension members.
3. Run the External General Ledger Data Loader program from the Corporate Performance Manager Integration Manager responsibility. Enter program parameters as described in the *Program Parameters* section of this document.
4. Monitor the concurrent request and view the log file when the execution is complete.

If the program terminated with a database error, correct the error and rerun the program with the same parameters.

If it completed with an error related to a processing restriction, review the *Data Load Execution Modes* and the *Processing Restrictions* sections of this document.

If you ran with some level of debug messaging enabled, you can view debug messages in the Debug Log (use the string 'FEM_%GL_POST%' in the *Module to view field* to see all messages from the External General Ledger Data Loader program).
5. If the process finishes with a warning that there are data errors, view the POSTING_ERROR_CODE column for each row in the interface table and refer to its description in the *Data Validation Errors* section in this document. Correct the error in each row by supplying the missing data, by correcting the invalid data, or by creating/loading missing dimension values in the Enterprise Performance Foundation database.

Repeat this process as needed to load data for each period. Remember that the first run for each period must be a Snapshot load. All subsequent runs for the same period must be Incremental loads. Each Incremental load run automatically attempts to reprocess rows previously rejected with data validation errors.

Detail Client Data Loader

Similar to other Oracle Enterprise Performance Foundation entities (such as dimension members and hierarchies), client data tables require a mechanism to populate their data from external systems. Each Detail Client Table Loader table has a different set of columns, and it offers a profile option for more detailed reporting of validation results on dimension columns.

The Client Data Tables

The Detail Client Data Loader supports loading data into client data tables such as the following:

- FEM_CHECKING
- FEM_COLLATERALS
- FEM_COMMERCIAL_LOANS

- FEM_CONSUMER_LOANS
- FEM_CREDIT_CARDS
- FEM_CREDIT_LIMITS
- FEM_CREDIT_LINES
- FEM_CUST_BOND_PRICES
- FEM_CUST_FINANCIAL_INFO
- FEM_CUST_SHARE_PRICES
- FEM_DATAx tables (FEM_DATA1 - FEM_DATA20)
- FEM_INVESTMENTS
- FEM_LEASES
- FEM_LOAN_LOSS
- FEM_MERCHANT_CARDS
- FEM_MORTGAGES
- FEM_MORTGAGE_BACK_SEC
- FEM_OFF_BALANCE_SHEET
- FEM_OTHER_SERVICES
- FEM_RETIREMENT_ACCOUNTS
- FEM_SAVINGS
- FEM_TERM_DEPOSITS
- FEM_TRADING_VAR
- FEM_TRUSTS
- FEM_WHOLESALE_FUNDING
- RCM_BANKING_MR
- RCM_IRB_RISK_COMPONENTS
- RCM_MASTER_NETTING_AGRMNT

Overview

The detail client data import process involves three main steps:

1. As a preparatory step, populate the dimension tables with all dimension members that are part of the input data. This includes population of Groups/Levels and possibly Hierarchy creation. (See related documentation for instructions on using the Dimension Member Loader and Dimension Hierarchy Loader.)
2. Populate the interface tables with data that is intended for the associated target tables, typically using SQL*Loader.
3. Run the Detail Client Data Loader program. This Loader reads data from the interface tables, performs some validation, and then imports valid data into the associated target tables or marks rows in the interface tables that have missing

required values or otherwise invalid data. Once all errors are resolved, the data load can be completed.

Populating the Interface Tables for the Detail Client Data Loader

The following section provides information on how to populate the Detail Client Data interface tables in preparation for a load into their associated target tables.

Detail Client Data Interface Tables

Each Detail Client Data table has a corresponding interface table. When you pick a Detail Client Data table to load, the Detail Client Data Loader loads only from the corresponding interface table. For example:

- FEM_CHECKING loads from FEM_CHECKING_T
- FEM_COLLATERALS loads from FEM_COLLATERALS_T
- FEM_COMMERCIAL_LOANS loads from FEM_COMMERCIAL_LOANS_T

Column Mapping

With the exception of the columns used to identify Calendar Period members, each column on the interface table maps to a single column on the target client data table. For example:

- GEOGRAPHY_DISPLAY_CODE maps to GEOGRAPHY_ID
- CCTR_ORG_DISPLAY_CODE maps to COMPANY_COST_CENTER_ORG_ID

The loader converts member Display Code identifiers into internal numeric identifiers during the load. Therefore, the numeric data values that appear in the target “_ID” column are not identical to the input Display Code values, which are typically alphanumeric. The numeric values are system generated. You will not need to map them in the interface tables. For reference only.

Dimension Columns

The following section provides examples of how to populate the various Dimension Columns, including those for the Calendar Period dimension, in preparation for loading. This is only a partial list.

Note: The Display Code values should match codes you have created through the Dimension Hierarchy Manager (such as for Ledger or Dataset).

- LEDGER_DISPLAY_CODE
Populate the LEDGER_DISPLAY_CODE column with a valid Ledger identifier. The Ledgers are identified in FEM_LEDGERS_VL.
- DATASET_DISPLAY_CODE
Populate the DATASET_DISPLAY_CODE column with a valid Dataset identifier. Datasets are identified in FEM_DATASETS_VL.
- SOURCE_SYSTEM_DISPLAY_CODE

Populate the SOURCE_SYSTEM_DISPLAY_CODE column with a valid Source System identifier. Source System members are identified in FEM_SOURCE_SYSTEMS_VL.

- CURRENCY_CODE

Populate the CURRENCY_CODE column with a valid Currency Code. Currency members are identified in FEM_CURRENCIES_VL.

- CAL_PERIOD_END_DATE, CAL_PERIOD_NUMBER, CALP_DIM_GRP_DISPLAY_CODE

The FEM Detail Client Data Loader requires the following for the identification of Calendar Period members. See the Introduction to this document for further details.

- CAL_PER_DIM_GRP_DISPLAY_CODE

This is the Level (Dimension Group) to which the Calendar Period belongs.

- CAL_PERIOD_END_DATE

This is the end date of the calendar period.

- CAL_PERIOD_NUMBER

This is the period number of the Calendar Period.

The following table contains columns for Calendar Period End Date, Level (Dimension Group Display Code), and Calendar Period Number:

CAL_PERIOD_END_DATE	CAL_PER_DIM_GRP_DIS PLAY_CODE	CAL_PERIOD_NUMBER
31-May-2004	Month	12
30-Jun-2004	Month	1
31-Jul-2004	Month	2
31-Aug-2004	Month	3
30-Sep-2004	Month	4
31-Oct-2004	Month	5
30-Nov-2004	Month	6
31-Dec-2004	Month	7
31-Jan-2005	Month	8
28-Feb-2005	Month	9
31-Mar-2005	Month	10
30-Apr-2005	Month	11
31-May-2005	Month	12
31-Aug-2004	Quarter	1
30-Nov-2004	Quarter	2
28-Feb-2005	Quarter	3
31-May-2005	Quarter	4
31-May-2005	Year	1
30-Jun-2005	Month	1

Because the loader uses the LEDGER_DISPLAY_CODE to obtain the Calendar for the time periods, the Calendar Periods of all rows in the <Detail Client Data>_T table for a given Ledger belong to the same Calendar.

- DISPLAY_CODE Columns for Value Set Dimensions

Other DISPLAY_CODE columns in the source <Detail Client Data>_T table are for Value Set dimensions. Populate each of these columns with a valid member Display Code from the appropriate dimension.

In a multi-value set implementation, the loader uses the LEDGER_DISPLAY_CODE to obtain the Value Set for validating the member Display Code values. The loader obtains the Value Set from the Global Value Set Combination associated to the Ledger.

Processing Key

The processing key is a set of columns on the target table that designates the dimensionality of the data for that table. You define the processing key for the target table by using the Administration application.

To ensure that there are no duplicate rows, you must create a corresponding unique index on the source table that has the same column list as the processing key for the target table. Because the loader uses a combination of columns on the source table to identify Calendar Period members while the target table has only a single column (CAL_PERIOD_ID), all of the Time columns on the source table need to be part of the unique index.

For example, assume that the data to be loaded has the following dimensionality:

- Calendar Period
- Channel
- Product

The processing key of the target table is therefore defined as:

- CAL_PERIOD_ID
- CHANNEL_ID
- PRODUCT_ID
- OBJECT_ID

Note: The processing key of the target table will probably include the OBJECT_ID column. Because there is no equivalent in the source interface table for this column, you can ignore it when you are constructing the unique index for the source interface table. When comparing the unique index of the source interface table, the loader ignores the existence of the OBJECT_ID column in the processing key.

The loader then requires that one unique index on the source table be defined as follows:

- LEDGER_DISPLAY_CODE
- CALP_DIM_GRP_DISPLAY_CODE
- CAL_PERIOD_END_DATE
- CAL_PERIOD_NUMBER
- CHANNEL_DISPLAY_CODE
- PRODUCT_DISPLAY_CODE

Running the Detail Client Data Loader

The purpose of the Detail Client Data Loader is to load data into a designated target table. The Detail Client Data Loader only inserts records. It does not update any records in the target table, nor does it allow incremental loads to the same Dataset and Calendar Period once there has been a successful *Snapshot* run without errors.

If the Loader encounters some data errors, the invalid records in the Interface table should be removed, or corrected and reloaded using the Error Reprocessing mode. When

using the Error Reprocessing mode, the Loader will attempt to load all appropriate rows of data where the status is not set to "LOAD".

If the first Snapshot ends Complete with no error, the data will be locked for that Dataset and Calendar Period. If you want to add more records, you must then remove the previous load and reload the full set of records. (The Executed Rule Removal program will unlock the data and remove the records from the target table. Then you may reload the interface table, and run the data loaders with Snapshot mode elected, since Status is set to LOAD on the interface records.)

Before You Launch the Loader

Verify that you have done the following before launching the Loader:

1. Dimensions values and, where appropriate, hierarchies were defined for Ledgers, Periods, Datasets and other dimensions. Once these values are loaded, you will need to perform this step only for new dimension members.

The Detail Client Data Loader requires that values for all Dimension members specified in the source interface table be loaded prior to loading the detail client data, as it uses the existing dimension member values during the validation process.

2. Data has been loaded into the interface (<Detail Client Data>_T) tables.
3. Your input parameters match the data in the <Detail Client Data>_T table(s) for the data you want to load:
 - Ledger = LEDGER_DISPLAY_CODE
 - Dataset = DATASET_DISPLAY_CODE
 - Source System = SOURCE_SYSTEM_DISPLAY_CODE
 - Calendar Period = the combination of LEDGER_DISPLAY_CODE, CALP_DIM_GRP_DISPLAY_CODE, CAL_PERIOD_END_DATE, and CAL_PERIOD_NUMBER.

Restrictions

The Detail Client Data Loader does not allow multiple users to load data into the same target <Detail Client Data> table at the same time. Users can simultaneously run the loader, however, as long as they are loading for separate target tables or using different load parameters (that is, different Ledgers, Calendar Periods, and so on).

Removing Loaded Data

You can remove loaded data from a client data table through the Schedule Requests pages (you cannot use the Executed Rules page to undo client data loader processes). For further information about working with requests, see About Concurrent Programs and Requests, page 8-1.

Concurrent Manager Program

Choose the *Detail Client Data Loader* concurrent program to launch the load.

Input Parameters

The Detail Client Data Loader requires selection of the following input parameters. As discussed earlier, these should be existing dimension values and should match the data you want to load:

- Ledger
- Dataset
- Source System
- Calendar Period

This consists of three values: `Cal_Per_Dim_Grp_Display_Code`, `Calendar_End_Date`, and `Cal_Period_Number`.

- Execution Mode

Pick an execution mode of either *Snapshot* or *Error Reprocessing* from the pick list. If you pick *Snapshot*, the loader loads all records where the `STATUS = 'LOAD'`. If you pick *Error Reprocessing*, the loader loads only those records where the `STATUS <> 'LOAD'`.

Logging

Concurrent Manager produces standard error and log message reports. In addition, you can set up debug messaging in System Administration Debug Logging on the User level. Oracle suggests Statement level logging for FEM% modules.

The Loader updates the `STATUS` column for error rows in the source interface tables with a code indicating what was wrong with the record. For example, the loader checks for required columns that are not populated, or invalid entries in dimension columns.

The level of detail of the error reporting during the dimension validation is determined by the FND profile “FEM: Loader Dimension Grouping Size”. This profile option can be set in System Administration Profile options for the User. *This option directly affects the processing speed.*

- Loader Dimension Grouping Size = default (currently null/blank in the profile options)

This will enable faster processing. When validating dimension values, however, the Loader updates the `STATUS` column with a *generic* error code (“FEM_SD_LDR_INV_DIM_MEMBR”) to let the user know that dimension validation failed for at least one dimension column, but it will not specify which column.

- Loader Dimension Grouping Size = 1

This may lengthen processing time. Using this setting, however, the loader will update the `STATUS` column of the interface table to indicate the *first dimension column that failed dimension validation* in the given row (see the “Processing and Validation Errors” section).

Output

Monitor the concurrent request and view the Concurrent Manager log file when the execution is complete. If you ran with some level of debug messaging enabled, you can view debug messages in the Debug Log.

When a load is successful, the Loader removes all rows for the detail client data from the various interface tables.

If the program terminated with a database error, correct the error and rerun the program with the same parameters.

If the process finishes with a warning that there are data errors, view the debug log. Also review the STATUS column for each row in the interface table and refer to its description in the *Processing and Data Validation Errors* section of this document. Correct the error in each row by supplying the missing data, correcting the invalid data, or by creating or loading missing dimension values into the database. Then run the Detail Client Data Loader in Error Reprocessing mode.

The following table lists the possible STATUS codes. When execution mode = *Snapshot*, records with a STATUS = 'LOAD' are loaded by the Detail Client Data Loader. (Note: This is case sensitive and must be in upper case.) All other STATUS codes designate some sort of error from the load.

Processing and Data Validation Errors

When a row in the interface table contains insufficient or invalid data, the Detail Client Data Loader populates the Status column for that row with one of these values. See also the "Logging" section regarding the profile option to set level of error detail written.

Processing Error or Interface row Status Code	Meaning
FEM_SD_LDR_DUPLICATE_ROW	This row along with any other row with this error may have violated the uniqueness constraint of the target table.
FEM_SD_LDR_INV_BAL_TYPE	Dataset code <DATASET> does not have a valid Balance Type attribute assignment. (This error should not occur. Call Customer Support.)
FEM_SD_LDR_INV_CAL_PER	Calendar Period ID <CAL_PER> or its display cannot be found, or it is not approved.
FEM_SD_LDR_INV_DATASET	Dataset code <DATASET> or its display code cannot be found, or it is not approved.
FEM_SD_LDR_INV_DIM_MEMBR	This row was not loaded because at least one of its dimension columns failed dimension validation. Used when Loader Dimension Grouping Size is default value (see the Logging section of this document).
FEM_SD_LDR_INV_EXEC_MODE	Execution mode must be either Snapshot ('S') or Error Reprocessing ('E'). (This error should not occur. Call Customer Support.)
FEM_SD_LDR_INV_LEDGER	Ledger ID <LEDGER> or its display code cannot be found, or it is not approved.
FEM_SD_LDR_INV_OBJ_DEF	Rule or its display code cannot be found, or it is not approved. (This error should not occur. Call Customer Support.)
FEM_SD_LDR_INV_SOURCE	System source code <SOURCE> or its display code cannot be found, or it is not approved.
FEM_SD_LDR_MISMATCH_CALENDAR	Invalid Ledger ID <LEDGER> or Calendar Period ID <CAL_PER> parameter. The calendar assigned to the ledger does not match the calendar assigned to the calendar period. (This error should not occur. Call Customer Support.)

Processing Error or Interface row Status Code	Meaning
FEM_SD_LDR_NO_SLICES	<p>The FEM Multiprocessing Framework was unable to generate any data slices with the program parameters. In most cases, this is due to no row in the interface table that matches with the parameters entered for this execution. It is also possible that the data slicing column or columns had NULL values. (Data slicing is set in the Schema Administration chapter, the Working with Process Tuning Options section.) The parameters entered are:</p> <ul style="list-style-type: none"> • CALP_DIM_GRP_DISPLAY_CODE: <DIM_GRP> • CAL_PERIOD_NUMBER: <PER_NUM> • CAL_PERIOD_END_DATE: <END_DATE> • LEDGER_DISPLAY_CODE: <LEDGER_DC> • DATASET_DISPLAY_CODE: <DATASET_DC> • SOURCE_SYSTEM_DISPLAY_CODE: <SOURCE_DC> <p>The execution mode is <EXEC_MODE>. If this is a Snapshot (S) run, only those rows where STATUS is 'LOAD' will be processed. If this is an Error Reprocessing (E) run, only those rows where STATUS is not 'LOAD' will be processed.</p>
FEM_SD_LDR_OBJ_DEF_INV_OBJ	<p>Rule <OBJ_DEF_ID> is not in the object catalog or is not an object type of 'SOURCE_DATA_LOADER'. (This error should not occur. Call Customer Support.)</p>
FEM_SD_LDR_PER_NOT_IN_LEDGER	<p>Invalid Ledger ID <LEDGER> or Calendar Period ID <CAL_PER> parameter. The calendar period is not in the calendar period hierarchy assigned to the ledger.</p>
FEM_SD_LDR_PROCESS_SUMMARY	<p><LOADNUM> rows loaded successfully. <REJECTNUM> rows failed to load successfully.</p>
FEM_SD_LDR_INV_DIM_MEMBER: <COLUMN NAME>	<p>Identifies the first dimension column that failed dimension validation in the current row. Used when Loader Dimension Grouping Size = 1 value (see Logging section of this document).</p>
LOAD	<p>The record is ready for loading.</p>

Table Descriptions

The Detail Client Data Loader can load any table that is registered in Enterprise Performance Foundation and has a corresponding interface table. The following list shows some of the tables supported by the Detail Client Data Loader:

Source Data (extracts created and inserted by customer)	Target Data (inserted by Detail Client Data Loader)
FEM_CHECKING_T	FEM_CHECKING
FEM_COMMERCIAL_LOANS_T	FEM_COMMERCIAL_LOANS
FEM_CONSUMER_LOANS_T	FEM_CONSUMER_LOANS
FEM_COLLATERALS_T	FEM_COLLATERALS
FEM_CREDIT_CARDS_T	FEM_CREDIT_CARDS
FEM_CREDIT_LIMITS_T	FEM_CREDIT_LIMITS
FEM_CREDIT_LINES_T	FEM_CREDIT_LINES
FEM_CUST_BOND_PRICES_T	FEM_CUST_BOND_PRICES
FEM_CUST_FINANCIAL_INFO_T	FEM_CUST_FINANCIAL_INFO
FEM_CUST_SHARE_PRICES_T	FEM_CUST_SHARE_PRICES
FEM_DATAx_T (FEM_DATA1_T - FEM_DATA20_T)	FEM_DATAx (FEM_DATA1 - FEM_DATA20)
FEM_INVESTMENTS_T	FEM_INVESTMENTS
FEM_LEASES_T	FEM_LEASES
FEM_LOAN_LOSS_T	FEM_LOAN_LOSS
FEM_MERCHANT_CARDS_T	FEM_MERCHANT_CARDS
FEM_MORTGAGE_BACK_SEC_T	FEM_MORTGAGE_BACK_SEC
FEM_MORTGAGES_T	FEM_MORTGAGES
FEM_OFF_BALANCE_SHEET_T	FEM_OFF_BALANCE_SHEET
FEM_OTHER_SERVICES_T	FEM_OTHER_SERVICES
FEM_RETIREMENT_ACCOUNTS_T	FEM_RETIREMENT_ACCOUNTS
FEM_SAVINGS_T	FEM_SAVINGS
FEM_TERM_DEPOSITS_T	FEM_TERM_DEPOSITS
FEM_TRADING_VAR_T	FEM_TRADING_VAR
FEM_TRUSTS_T	FEM_TRUSTS
FEM_WHOLESALE_FUNDING_T	FEM_WHOLESALE_FUNDING
RCM_BANKING_MR_T	RCM_BANKING_MR
RCM_IRB_RISK_COMPONENTS_T	RCM_IRB_RISK_COMPONENTS
RCM_MASTER_NETTING_AGRMNT_T	RCM_MASTER_NETTING_AGRMNT

FEM_DATAx_T

This interface table stores data for import into Enterprise Performance Foundation. There are 20 of these tables, one for each FEM_DATAx table in the system.

Example names for these interface tables are as follows:

- FEM_DATA1_T
- FEM_DATA2_T
- FEM_DATA20_T

Column Name	Datatype	Nullable	Column Description
DATASET_CODE	NUMBER	Not Null	Represents the data set for the balance record. This is one component of the primary key and foreign key to the Data Set dimension.
CALP_DIM_GRP_DISPLAY_CODE	VARCHAR2(150)	Not Null	Designates the Dimension Group Display Code of the Calendar Period. This is used in conjunction with the CAL_PERIOD_END_DATE, CAL_PERIOD_NUMBER, and LEDGER_ID to determine the CAL_PERIOD_ID for the row when it is loaded into FEM. Note that the Ledger ID provides the information to get the Calendar ID. Knowing the Calendar ID, End Date, Dimension Group, and Cal Period Number allows a look up of the CAL_PERIOD_ID.
CAL_PERIOD_END_DATE	DATE	Not Null	Designates the End Date for the Period. This is used in conjunction with the CAL_PERIOD_NUMBER, CALP_DIM_GRP_DISPLAY_CODE, and LEDGER_DISPLAY_CODE to identify the CAL_PERIOD_ID for the row when it is loaded.

Column Name	Datatype	Nullable	Column Description
CAL_PERIOD_NUMBER	NUMBER(15)	Not Null	Designates the Period Number of the Calendar Period (example – 1 for JAN, 2 for MARCH, 1 for first Quarter, and so on). This is used in conjunction with the CAL_PERIOD_END_DATE, CAL_P_DIM_GRP_DISPLAY_CODE, and LEDGER_DISPLAY_CODE to determine the CAL_PERIOD_ID for the row when it is loaded into Enterprise Performance Foundation.
SOURCE_SYSTEM_CODE	NUMBER	Not Null	Foreign key to the Source System dimension.
LEDGER_DISPLAY_CODE	VARCHAR2(150)	Not Null	Foreign key to the GL Set of Books dimension
CCTR_ORG_DISPLAY_CODE	VARCHAR2(150)		Foreign key to the Organization dimension
CURRENCY_CODE	VARCHAR2(15)		Foreign key to the Currency dimension.
FINANCIAL_ELEMENT_DISPLAY_CODE	VARCHAR2(150)		Foreign key to the Financial Element dimension
PRODUCT_DISPLAY_CODE	VARCHAR2(150)		Foreign key to the Product dimension
NATURAL_ACCOUNT_DISPLAY_CODE	VARCHAR2(150)		Foreign key to the Natural Account dimension.
CHANNEL_DISPLAY_CODE	VARCHAR2(150)		Foreign key to the Channel dimension.
LINE_ITEM_DISPLAY_CODE	VARCHAR2(150)		Foreign key to the Line Item dimension
PROJECT_DISPLAY_CODE	VARCHAR2(150)		Foreign key to the Project dimension.
CUSTOMER_DISPLAY_CODE	VARCHAR2(150)		Foreign key to Customer dimension

Column Name	Datatype	Nullable	Column Description
INTERCOMPANY_ DISPLAY_CODE	VARCHAR2(150)		Foreign key to the Company Cost Center Organization dimension (GCSII).
ENTITY_ DISPLAY_ CODE	VARCHAR2(150)		Foreign key to the Entity dimension (GCSII).
GEOGRAPHY_DIS PLAY_CODE	VARCHAR2(150)		Foreign key to the Geography dimension.
TASK_DISPLAY_ CODE	VARCHAR2(150)		Foreign key to the Task dimension.
USER_DIM1_ DIS PLAY_CODE	VARCHAR2(150)		Foreign key to user-configurable dimension.
USER_DIM2_ DIS PLAY_CODE	VARCHAR2(150)		Foreign key to user-configurable dimension.
USER_DIM3_ DIS PLAY_CODE	VARCHAR2(150)		Foreign key to user-configurable dimension.
USER_DIM4_ DIS PLAY_CODE	VARCHAR2(150)		Foreign key to user-configurable dimension.
USER_DIM5_ DIS PLAY_CODE	VARCHAR2(150)		Foreign key to user-configurable dimension.
USER_DIM6_ DIS PLAY_CODE	VARCHAR2(150)		Foreign key to user-configurable dimension.
USER_DIM7_ DIS PLAY_CODE	VARCHAR2(150)		Foreign key to user-configurable dimension.
USER_DIM8_ DIS PLAY_CODE	VARCHAR2(150)		Foreign key to user-configurable dimension.
USER_DIM9_ DIS PLAY_CODE	VARCHAR2(150)		Foreign key to user-configurable dimension.
USER_DIM10_ DIS PLAY_CODE	VARCHAR2(150)		Foreign key to user-configurable dimension.
NUMERIC_ MEASURE	NUMBER		Generic attribute column for numeric data.

Column Name	Datatype	Nullable	Column Description
ALPHANUMERIC_MEASURE	VARCHAR2(100)		Generic column for alphanumeric data.
DATE_MEASURE	DATE		Generic column for date data.
STATUS	VARCHAR2(50)	Not Null	Designates the Status of the row for loading. Records with a STATUS = 'LOAD' are loaded by the Enterprise Performance Foundation DataX Loader. All other STATUS codes designate some sort of error from the load.

The primary key is defined by the user at implementation. It is identical to the primary key of the corresponding FEM_DATAx table.

Sample SQL*Loader DAT and CTL Files for Detail Client Data Interface Tables

The following section provides examples of SQL*Loader DAT and CTL files for loading into the Detail Client Data interface tables.

Detail Client Data Interface Table(s)

- CTL file for FEM_COMMERCIAL_LOANS_T

```

OPTIONS (ERRORS=0, SILENT=(FEEDBACK), DIRECT=TRUE)
LOAD DATA
INFILE 'FEM_COMMERCIAL_LOANS_T.dat'
BADFILE ' FEM_COMMERCIAL_LOANS_T.bad'
DISCARDFILE ' FEM_COMMERCIAL_LOANS_T.dsc'
INTO TABLE FEM_COMMERCIAL_LOANS_T
REPLACE
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
(
  DATASET_DISPLAY_CODE,
  CAL_PERIOD_END_DATE,
  CAL_PERIOD_NUMBER,
  CALP_DIM_GRP_DISPLAY_CODE,
  SOURCE_SYSTEM_DISPLAY_CODE,
  LEDGER_DISPLAY_CODE,
  CURRENCY_CODE,
  LINE_ITEM_DISPLAY_CODE,
  CCTR_ORG_DISPLAY_CODE,
  .
  . (list all fields)
  .
  STATUS

```

- DAT file for FEM_COMMERCIAL_LOANS_T

```
"ACTUAL", "31-MAR-2004", "3", "CAL_PERIOD_L3", "EPB", "EPBLEDDGER",  
"EUR", "LN0028", "M78", ..., "LOAD"  
"ACTUAL", "31-MAR-2004", "3", "CAL_PERIOD_L3", "EPB", "EPBLEDDGER",  
"USD", "LN0028", "M78", ..., "LOAD"
```

Replacement Mode for the Detail Client Data Table Loader

This mode provides the ability to add and replace data previously loaded in a data table for the same ledger, calendar period, dataset, and source system combination, without requiring an undo of the earlier load.

The details of the Replacement execution mode are as follows:

- Adds the “Replacement” mode to the “Snapshot” and “Error Reprocessing” loader modes.
- Only available for the Detail Client Data Loader concurrent program. This provides the ability to add and replace data in FEM_DATA1-20 or any client data table.
- Replacement mode requires an initial Snapshot data load for a given ledger, calendar period, dataset and source system code combination
- Provides the ability to add new data or to update previously loaded data for the same ledger, calendar period, dataset and source system code without requiring an undo operation on previously loaded client data.
- Based on the client data table’s processing key, it replaces existing rows having the same key values, and inserts any rows having new key values – no other rows are affected.
- When reading the interface table, it ignores the Status column. (In comparison: Snapshot mode loads rows where the Status is set to “LOAD”, and Error Reprocessing mode loads rows where the Status indicates an error – i.e., \diamond “LOAD”.) Consistent with the other modes, Replacement validates display code entries and writes an error to the Status column if it encounters an invalid display code.
- Upon successful completion, business event oracle.apps.fem.loader.fact.execute is raised. The user must subscribe to this business event if they want to be notified when the “Replacement” process has completed. Please reference the Oracle Workflow Guide for complete details related to business events and subscriptions.
- To execute a client data table load in Replacement mode, select “Replacement” as the execution mode parameter when submitting the “Detail Client Data Loader” concurrent process.

Dimension Loader

In the Enterprise Performance Foundation data model, users generally migrate Dimensions and Hierarchies from an external source. This data is initially imported into the interface tables and then, after validations, it is moved to the Enterprise Performance Foundation core metadata tables where it can be used for processing and analysis.

This feature allows dimension loading rules to be created as stored rules with scope and run any time with or without modifications.

Business Process

The Business Process related to the Loader Rules feature is as follows:

1. Create Loader Rules for Dimensions and or Hierarchies
2. View, modify, or delete Loader Rules
3. Run the Loader Rules

You can make changes and rerun the rules as many times as required.

Accessing the Dimension Loader Rule feature

To access the Dimension Loader Rules feature:

1. Select the Enterprise Performance Foundation Administrator Responsibility
2. Select the **Dimension Rules** option
3. Select **Dimension Loader** on the submenu

Creating a Dimension Loader Rule

To create new Loaders Rule for Dimensions or Hierarchies, click **Create** on the Rules home page, page 2-2

Step 1 – Specify Dimension Loader Rule Details

The following table describes the **Create Dimension Loader Rule Detail** page components:

Create Dimension Loader Rule Detail Page Components

Component	Description
Folder Name	(Mandatory) The default security folder is specified in Application Preference, you can also select any other available folder.
Rule Name	(Mandatory) Specify the name of the rule.
Description	(Optional) We recommend a detailed description of the rule.
Access	Read and Write - The rule can be modified Read Only – The Rule can only be viewed.

Step 2 – Specify Dimension Loader Rule Version Details

The following table describes the **Create Dimension Loader: Version** page components

Create Dimension Loader Rule Version Details Page Component

Component	Description
Folder Name	(Mandatory) The default security folder is specified in Application Preference, you can also select any other available folder.
Rule Name	(Mandatory) Specify the name of the rule.
Description	(Optional) We recommend a detailed description of the rule.
Access	Read and Write - The rule can be modified Read Only – The Rule can only be viewed.

Step 3 – Specify Dimension Parameter Details

The tool at the top of the **Create Dimension Loader: Dimensions** page lets you perform simple queries on dimensions based on their Value Set status. The following table describes the other components on this page.

Create Dimension Loader Rule Dimension Parameter Details Page Component

Component	Description
Value set Required	Yes - The list box shows only dimensions which are of the type Value Set Required No - The list box shows only those Dimensions which are of the type Value set not required. Blank - The list box shows all the dimensions irrespective of their Value set
Dimension Name	Use this text box to query a specific dimensions by entering all or part of a dimension name.
Move	Move a particular dimension in the “Available Dimensions” box to the “Selected Dimensions” box
Move All	Move all dimensions in the “Available Dimensions” box to the “Selected Dimensions” box
Remove	Move a particular dimension on the “Selected Dimensions” box to the “Available Dimensions” box
Remove All	Move all the dimensions on the “Selected Dimensions” box to the “Available Dimensions” box
Selected Dimensions	This box shows all the dimensions that have been selected for this rule definition.

Step 4: – Hierarchy Parameter Details

Create Dimension Loader Rule Hierarchies Page Component

Component	Description
Auto Populate Dimensions	The selected dimensions will be auto populated for hierarchy loading. You can modify, add, or delete rows. If you add a parameter that has already been specified in this page, it will be ignored.
Add Another Row	An empty row will be created.
Dimension	The name of the dimension on which the hierarchy to be loaded was.
Hierarchy	Specify the hierarchy name, or specify All to load every hierarchy in the interface tables for the selected dimension.
Version	Specify the version of the hierarchy or specify All to load all the versions for the hierarchy.
Duplicate	Populate a new record with the contents of the selected row.
Delete	Remove a row

Duplicating Dimension Loader Rules

You can use a rule's existing version to create additional versions or you can create a new rule and version by clicking the **Duplicate** button for a Rule's version, which takes you to the "Step1: Duplicate Options" page

For more information, refer to Common Rule Management Tasks, page 2-1.

Executing Dimension Loader Rules

It is important to note that before executing the Dimension Loader Rule, you must have populated records into the respective Interface Tables to enable the rule to successfully fetch them for loading into the EPF Core Tables.

1. On the Rules home page, page 2-2, expand the Rule to see all the Rule Versions under it
2. Select a rule version
3. Click **Run**.

The Execute Rule page is displayed. Enter information as required

Component	Description
Rule Name	The Rule selected for execution
Rule Start Date	The selected version start date
Rule End Date	The selected version end date
Submit	Executes the rule. You will be taken to a screen displaying the concurrent request number. Make a note of the number and proceed to "View Requests" to see the progress of the execution. On successful completion, review the related log files for the status of various loadings initiated by the rule execution.

Data Loader

In the Enterprise Performance Foundation data model, users generally migrate Account, Ledger and Client Data from an external source. This data is initially imported into the interface tables and then, after validations, it is moved to the EPF core metadata tables where it can be used for processing and analysis.

This feature allows Account, Ledger and Client data loading rules to be created as stored rules with scope and run any time with or without modifications.

Business Process

The Business Process related to the Data Loader Rules feature is:

1. Create Loader Rules for Account, Ledger, or Client Data
2. View, modify, or delete Loader Rules
3. Run the Loader Rules

You can make changes and rerun the rules as many times as required.

Accessing the Data Loader Rule feature

To access the data Loader rules feature:

1. Select the Enterprise Performance Foundation Administrator Responsibility
2. Select the **Data Rules** option
3. Select **Data Loader** on the submenu

Creating a Data Loader Rule

To create new Loader Rule for either Client Data or Ledger Data, click **Create** on the Data Loader Rules home page, page 2-2.

Step 1 - Specify Data Loader Rule Details

Create Data Loader Rule Details Page Components

Component	Description
Folder Name	(Mandatory) The default security folder is specified in Application Preference, you can also select any other available folder.
Rule Name	(Mandatory) Specify the name of the rule.
Description	(Optional) We recommend a detailed description of the rule.
Access	<ul style="list-style-type: none">• Read and Write - The rule can be modified• Read Only – The Rule can only be viewed.
Load Type	<ul style="list-style-type: none">• Client Data Load - The rule loads from Client Data Interface Tables• Ledger Data Load - The rule loads from Ledger Data Interface Tables

Step 2 - Specify Data Loader Rule Version Details

Create Data Loader Rule Version Details Page Components

Component	Description
Version	(Mandatory) - Specify the Rule version name.
Description	(Optional) - Oracle recommends a detailed description of the rule version.
Start Date	(Mandatory) - Specify the version's validity start date.
End Date	(Mandatory) - Specify the version's validity end date.

Step 3 – Rule Parameter Details

There are two options depending on the Load Type specified in Step 1.

Option 1 – Load Type set to Ledger Data Load

Create Data Loader Parameters Page Components

Component	Description
Add Another Row	There are no active rows for selecting parameters until this button is pressed.
Ledger	Specify the Ledger from which records are loaded during the execution of this rule. To specify all the Ledgers, select All or use button next to the field to select the desired Ledger through from a popup window.
Data Set	Specify the Dataset to be assigned in the EPF Core Table FEM_BALANCES when they are loaded from the Interface table FEM_BAL_INTERFACE_T. Use the button next to the field to select the desired Dataset from a popup window.
Load Option	<p>Data (Default) - When this rule is executed, only the records from the Interface table of Ledger (FEM_BAL_INTERFACE_T) are considered. Dimension or hierarchy interface table records are not loaded.</p> <p>Data and Dimension - When this rule is executed, the records from Interface tables of Client Data or Ledger is considered in addition to records of Interface tables of all the Dimensions of the Data Table Columns.</p> <p>Data, Dimension and Hierarchy - 'When this rule is executed the records from Interface tables of Client Data or Ledger is considered in addition to records of Interface tables of all the Dimensions of the Data Table Columns and their Hierarchy Interface Table records.</p>
Period Level	Specify the Period Level parameter to select records from the Interface table during the execution of this rule. To consider all the Period levels for loading, specify All. Please note that when a specific Ledger is selected in the Ledger field then only period levels related to the Calendar of that that Ledger would be available for selection (in addition to All). If All is specified in the Ledger parameter field then Period Level would show only All value for the selection. Use the button next to the field to select desired period level from a popup window.

Option 2 – Load Type set to Client Data Load

Create Data Loader Parameters Page Components

Component	Description
Add Another Row	There are no active rows for selecting parameters until this button is pressed.
Data Table Name	Specify the Client Data Table Name. Use the button next to the field to select the desired client data table from a popup window.
Source System	Specify the Source System to select records from the Interface table during the execution of this rule. To consider all the source systems, select All. Use the button next to the field to select the source system from a popup window.
Ledger	Specify the Ledger from which records are loaded during the execution of this rule. To specify all the Ledgers, select All or use button next to the field to select the desired Ledger through from a popup window.
Data Set	Specify the Dataset to select records from the Interface table during the execution of this rule. To specify all the Ledgers, select All or use button next to the field to select the desired Dataset through from a popup window.
Load Option	<p>Data (Default) - When this rule is executed, Only the Interface table of Client Data are considered. Dimension or hierarchy interface table records are not loaded.</p> <p>Data and Dimension - When this rule is executed, the records from Interface tables of Client Data or Ledger is considered in addition to records of Interface tables of all the Dimensions of the Data Table Columns.</p> <p>Data, Dimension and Hierarchy - 'When this rule is executed the records from Interface tables of Client Data or Ledger is considered in addition to records of Interface tables of all the Dimensions of the Data Table Columns and their Hierarchy Interface Table records.</p>
Period Level	Please note that when a specific Ledger is selected in the Ledger field then only period levels related to the Calendar of that Ledger would be available for selection (in addition to All). If All is specified in the Ledger parameter field then Period Level would show only All value for the selection. Use the button next to the field to select desired period level from a popup window.

Duplicating Data Loader Rules

You can use a rule's existing version to create additional versions or you can create a new rule and version by clicking **Duplicate** for a Rule's version, which takes you to the "Step1: Duplicate Options" page

For more information, refer to Common Rule Management Tasks, page 2-1.

Executing Data Loader Rules

It is important to note that before executing the Dimension Loader Rule, you must have populated records into the respective Interface Tables to enable the rule to successfully fetch them for loading into the EPF Core Tables.

1. On the Rule home page, page 2-2, expand the Rule to see all the Rule Versions under it
2. Select a rule version
3. Click **Run**.

The Execute Rule page is displayed. Enter information as required

Component	Description
Rule Name	The Rule selected for execution
Rule Start Date	The selected version start date
Rule End Date	The selected version end date
Data Selection Start Date	Enter the start date for records in the Interface tables to match with the column CAL_PERIOD_END_DATE
Data Selection End Date	<p>Enter the end date for records in the Interface tables to match with the column CAL_PERIOD_END_DATE</p> <p>Note: Only the records in Interface Tables falling within Start and End Date by matching with CAL_PERIOD_END_DATE would be loaded (in addition to matching other parameters specified in the rule).</p>
Balance Type	<p>Only mandatory if the selected Rule Version has the Ledger Data Load option selected.</p> <p>Select from the choice list whether the loading data will populate records in EPF tables for balance types PTD alone or along with, QTD and or YTD.</p>
Submit	<p>Executes the rule. You will be taken to a screen displaying the concurrent request number. Make a note of the number and proceed to "View Requests" to see the progress of the execution.</p> <p>On successful completion, review the related log files for the status of various loadings initiated by the rule execution.</p> <p>Refer to the <i>Oracle Enterprise Performance Foundation User's Guide</i> for information on the errors and status of rule execution.</p>

Profile Options

This appendix covers the following topics:

- Enterprise Performance Foundation Profile Options
- Profile Options in Application Preferences

Enterprise Performance Foundation Profile Options

The following table lists the profile options for Enterprise Performance Foundation.

Profile Name	Profile Description
FEM: Application Code	Displays the application defined for the user.
FEM: Attribute Interface Table Date Format Mask	Currently, when users enter start and end dates for calendar periods with the user interface, they do not allowed enter the "time" portion of the date, and this always is defaulted in the database to 00:00:00. However, in some situations, users may need to create what is called an "intra day" period. This is the case where a start time or end time represents less than a whole day. To do this, you must specify the "date" and the "time" for that period, and this is provided as a feature in the dimension member loader. To create an "intraday" period, the user must specify the date format mask in the "FEM: Attribute Interface Table Date Format Mask" profile option, and load the calendar period members into the appropriate loader table, and then run the dimension member loader program. By default, this profile option is set to exclude the time portion.
FEM: Currency Conversion Type	Used to specify the currency conversion type when translating currencies. The domain for this profile option is defined in the E-Business Suite, and includes options such as Corporate, Daily, and Spot.
FEM: Currency Type	The domain for this profile option is either Entered or Translated. Profitability Manager processes only those rows indicated as Entered. Rows indicated as Translated are primarily for reporting purposes.

Profile Name	Profile Description
FEM: Default Actuals Dataset	Specifies the default actuals data set used when setting up General Ledger Balances Rules to leverage actual balances in Enterprise Performance Foundation.
FEM: Default Effective Start Date	Specifies the start date default for rule versions. If not specified, the system will default to 1/1/1900.
FEM: Default Effective End Date	Specifies the end date default for rule versions. If not specified, the system will default to 1/1/2500.
FEM: FEM Top	Specifies the location of the FEM_TOP file system directory in the Oracle Applications file system. Used by the Enterprise Performance Foundation Refresh Engine.
FEM: GL Advanced Mapping Flag	Determines if line items and financial elements are associated with natural accounts when leveraging balances from Oracle General Ledger in Enterprise Performance Foundation. When the profile option is set to Yes, users should set up the Financial Element and Line Item attributes for the Natural Account dimension.
FEM: Loader Dimension Grouping Size	Number of dimensions to process at a time when validating the dimensions in the interface data. See the "Logging" topic under Running the Detail Client Data Loader, page A-93 for further information.
FEM: Rule Migration Access	Set by the system administrator to enable or disable migration functionality. If the profile option value is set to "N", the Migrate icon is disabled on the rule home page; otherwise, the icon is enabled. Can be set at the Site, Application, and Responsibility levels, but not at the User level.
FEM: Rule Set Depth Limit	Controls the depth of the nested rule sets within a rule set.
FEM: Signage Methodology	Identifies the signage methodology used for calculations that write to the FEM_BALANCES table and data written to Enterprise Performance Foundation as part of Oracle General Ledger integration. The available methodologies are: <ul style="list-style-type: none"> • Absolute Value - Loads the account types Assets and Expenses with no signage adjustment, but reverses signs for all balances with the account types Equity, Liabilities, and Revenue • Standard General Accounting Principles - Makes no signage adjustment • Reverse General Accounting Principles - Reverses signs for all balances

Profile Name	Profile Description
FEM: Track Event Chains	Supports the ability to undo a chain of processing runs. If this flag is set, and you undo a rule, any other rules that are dependent upon the undone rule are also undone.
FND: View Object Max fetch size	Sets the maximum number of rows displayed in lists of values and on pages in the user interface by restricting the number of rows that the view object fetches from the database. The default value is 200.

Profile Options in Application Preferences

Certain profile options are set by each user in Application Preferences within the application. These profile options should be set or updated only by the individual user through Application Preferences, and are as follows:

- FEM: Data Set Group
- FEM: Dataset
- FEM: Effective Date
- FEM: FEM Top
- FEM: Ledger
- FEM: Period
- FEM: Security Folder

Using Oracle Enterprise Performance Foundation with Oracle Discoverer

This appendix covers the following topics:

- Using Oracle Enterprise Performance Foundation with Oracle Discoverer

Using Oracle Enterprise Performance Foundation with Oracle Discoverer

The "EPF - Core Business Area" is an Oracle Discoverer business area that is packaged with Oracle Enterprise Performance Foundation to facilitate reporting against the data within the system. To use this business area, import it into the appropriate end user layer (EUL) for your specific installation.

Oracle Discoverer workbooks are included with individual product patches. Import the workbooks into the same EUL where you have imported the business area.

Instructions for the installation of the "EPF Core Business Area" are included in the *Oracle Enterprise Performance Foundation About Document*, and details related to the product-specific workbooks are included in the product-specific About Documents and User's Guides.

Business Events

Business Events

The following table lists the business events in Enterprise Performance Foundation.

Business Event	Description
oracle.apps.fem.dhm.dimension.event	Generated for each new dimension member added to Enterprise Performance Foundation through the Dimension and Hierarchy Management facility.
oracle.apps.fem.dhm.hierarchy.event	Generated for each new hierarchy added to Enterprise Performance Foundation through the Dimension and Hierarchy Management facility.
oracle.apps.fem.prockey.updated	Generated each time a user changes the processing key on a table.
oracle.apps.fem.ud.complete	Generated each time an undo process completes.
oracle.apps.fem.ud.submit	Generated each time an undo process is submitted to the concurrent manager.
oracle.apps.fem.oglintg.balrule.execute	Generated each time an Oracle General Ledger balance rule is submitted for processing.
oracle.apps.fem.xglintg.balrule.execute	Generated each time an external ledger integration process is submitted for processing.

Integrating Enterprise Performance Foundation with Oracle Warehouse Builder

This appendix covers the following topics:

- Introduction
- Installation
- Integration Components
- User Procedures

Introduction

This appendix describes how to set up and employ Oracle Warehouse Builder with Enterprise Performance Foundation.

Oracle Warehouse Builder is a comprehensive tool for practitioners who move and transform data, develop and implement business intelligence systems, perform metadata management, or create and manage Oracle databases and metadata.

Installation

Prerequisites

1. Install Oracle Warehouse Builder 10gR2 on the same database instance where Oracle E-Business Suite 11i is installed. Please refer to the *Oracle WarehouseBuilder 10gR2 Installation Guide* for installation information.
2. Create the following users:
 - Oracle Warehouse Builder Repository Owner: OWBREP
 - Oracle Warehouse Builder Repository User: OWBDPL

Patch details and Installation Procedure

The patch contains the following files:

- MDL file (FEMINTFLOAD.mdl) – contains the Oracle Warehouse Builder metadata for all the interface tables, staging tables, mappings and the Enterprise Performance Foundation Install Expert.

- SQL file (FEMOWBPRIVS.sql) – contains commands to create various grants and synonyms for the tables involved.

These files should be copied to a location from which the Oracle Warehouse Builder Design Center client and SQL*Plus can be accessed. Since ADPATCH currently does not support Oracle Warehouse Builder related files the files should be applied manually and not using ADPATCH.

1. Using SQL*Plus connect to the database using APPS account and run the FEMOWBPRIVS.sql file.
2. Start the Oracle Warehouse Builder Design Center and connect to the appropriate database and import the MDL file into the repository.

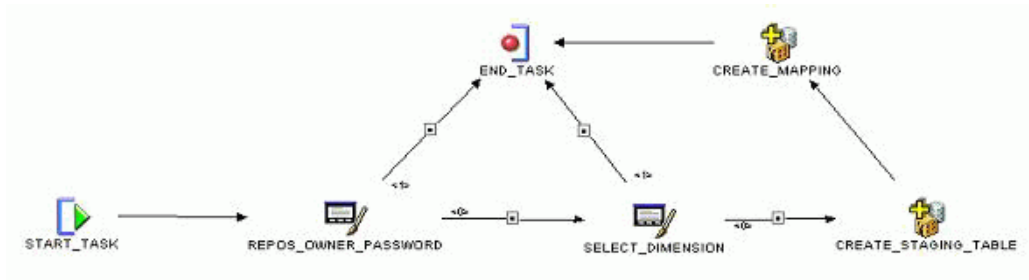
Integration Components

Enterprise Performance Foundation Interface Tables:

All the Enterprise Performance Foundation Interface tables are exposed as Oracle Warehouse Builder object so that mappings can be created various Staging tables and any external tables if necessary.

Enterprise Performance Foundation Install Expert

A wizard based navigation (called Enterprise Performance Foundation Install Expert) using Oracle Warehouse Builder Expert technology is created to refresh the Staging tables and Mappings if dimensional metadata changes.



GRAPHIC

Simplified Staging tables:

A simplified Staging table is provided for each of the Dimensions supported in this feature. The Staging table contains a column for each of the Attribute of the Dimension in addition to Member related columns. The following is the list of various flavors of Staging tables provided:

FEM_<xdim>_STG (for Value Set constrained Dimensions)

The following table describes the table components.

Column Name	Data Type	Size	Not Null	PK	Column Description
<xdim>_NAME	VARCHAR2	150	Y		Member Name
<xdim>_DISPLAY_CODE	VARCHAR2	150	Y		Member Display Code
DESCRIPTION	VARCHAR2	255			Description
VALUE_SET_DISPLAY_CODE	VARCHAR2	150	Y		Value Set Display Code
DIMENSION_GROUP_DISPLAY_CODE	VARCHAR2	150			Dimension Group Display Code
<required_attribute_name_1>	VARCHAR2	30			Attribute Name 1
<required_attribute_name_2>	VARCHAR2				Attribute Name 2
..					
<required_attribute_name_n>	VARCHAR2	30			Attribute Name n
<user_defined_attribute_name_1>	VARCHAR2	30			Attribute Name 1
<user_defined_attribute_name_2>	VARCHAR2	30			Attribute Name 2
..					
<user_defined_attribute_name_n>	VARCHAR2	30			Attribute Name n

FEM_<xdim>_STG (for non-Value Set constrained Dimensions)

The following table describes the table components.

Column Name	Data Type	Size	Not Null	PK	Column Description
<xdim>_NAME	VARCHAR2	150	Y		Member Name
<xdim>_DISPLAY_CODE	VARCHAR2	150	Y		Member Display Code
DESCRIPTION	VARCHAR2	255			Description
DIMENSION_GROUP_DISPLAY_CODE	VARCHAR2	150			Dimension Group Display Code
<required_attribute_name_1>	VARCHAR2	30			Attribute Name 1
<required_attribute_name_2>	VARCHAR2				Attribute Name 2
..					
<required_attribute_name_n>	VARCHAR2	30			Attribute Name n
<user_defined_attribute_name_1>	VARCHAR2	30			Attribute Name 1
<user_defined_attribute_name_2>	VARCHAR2	30			Attribute Name 2
<user_defined_attribute_name_n>	VARCHAR2	30			Attribute Name n

FEM_CAL_PERIODS_STG

The following table describes the table components.

Column Name	Data Type	Size	Not Null	PK	Column Description
CAL_PERIOD_END_DATE	DATE				Calendar Period End Date
CAL_PERIOD_NUMBER	NUMBER	15			Calendar Period Number
CALENDAR_DISPLAY_CODE	DATE	11			Calendar Display Code
VALUE_SET_DISPLAY_CODE	VARCHAR2	150			Value Set Display Code
DIMENSION_GROUP_DISPLAY_CODE	VARCHAR2	150			Dimension Display Code
<required_attribute_name_1>	VARCHAR2	30			Attribute Name 1
<required_attribute_name_2>	VARCHAR2	30			Attribute Name 2
..					
<required_attribute_name_n>	VARCHAR2	30			Attribute Name n
<user_defined_attribute_name_1>	VARCHAR2	30			Attribute Name 1

FEM_LEDGERS_STG

The following table describes the table components.

Column Name	Data Type	Size	Not Null	PK	Column Description
LEDGER_NAME	VARCHAR2	150			Ledger Name
LEDGER_DISPLAY_CODE	VARCHAR2	150			Ledger Display Code
DESCRIPTION	VARCHAR2	255			Description
<required_attribute_name_1>	VARCHAR2	30			Attribute Name 1
<required_attribute_name_2>	VARCHAR2	30			Attribute Name 2
..					
<required_attribute_name_n>	VARCHAR2	30			Attribute Name n
<user_defined_attribute_name_1>	VARCHAR2	30			Attribute Name 1
<user_defined_attribute_name_2>	VARCHAR2	30			Attribute Name 2
<user_defined_attribute_name_n>	VARCHAR2	30			Attribute Name n

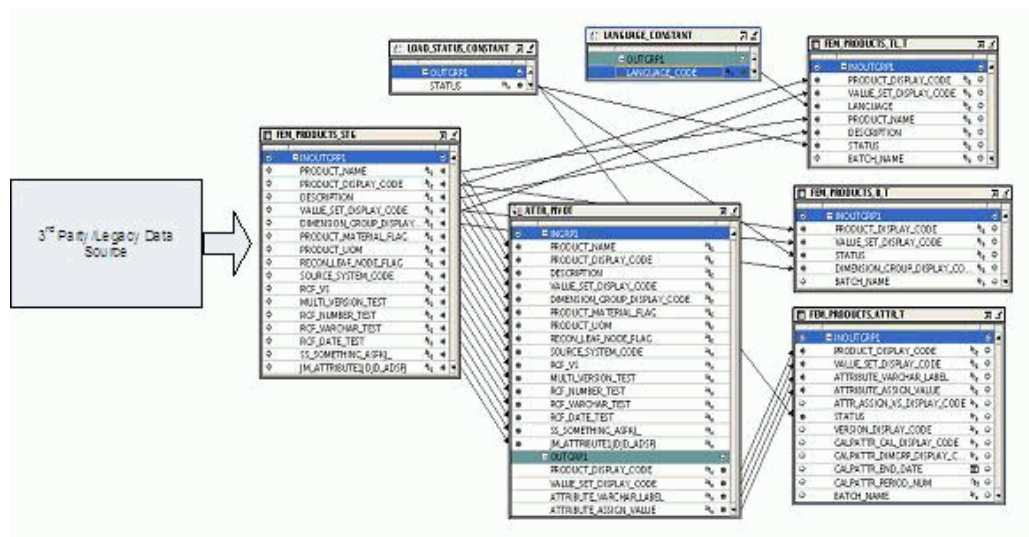
FEM_SIMPLE_DIMS_STG

The staging table for the Simple Dimensions (User List of Values Dimensions) will not have attribute columns, as these Dimensions do not support attributes. Even though the loading of data for Dimensions is fairly straight forward, we do expose the interface tables through Oracle Warehouse Builder and create staging table and mapping.

Column Name	Data Type	Size	Not Null	PK	Column Description
DIMENSION_VARCHAR_LABEL	VARCHAR2	30			Dimension Varchar Label
MEMBER_CODE	VARCHAR2	30			Member Display Code
MEMBER_NAME	VARCHAR2	150			Member Name
DESCRIPTION	VARCHAR2	255			Description

Preconfigured Mappings:

Mappings that move data from the Staging table to the Member and Attribute tables (if applicable) for each Dimension in Enterprise Performance Foundation are provided. The following is a sample Mapping provide:



User Procedures

Importing Dimension Member data into Enterprise Performance Foundation

Use the following steps to import dimension member data into Enterprise Performance Foundation

Step I: Refresh Repository:

If new dimension attributes have been created in Enterprise Performance Foundation and user wish to include them in the load process, user needs to run "Expert" from the Oracle Warehouse Builder Design Center to refresh the Staging table and Mapping for the Dimensions in question. The use of Enterprise Performance Foundation Install Expert will let the user choose one or more Dimensions for which the staging table and

mapping has to be created and once all the relevant information is gathered the mapping that directs the flow of data from the staging table to the interface tables will be created.

Step II: Deploy Staging tables and Mappings:

Use Oracle Warehouse Builder Control Center Manager and deploy Enterprise Performance Foundation Staging tables and Mappings on the default location called OWBDPL_LOCATION. This target location is nothing but the deployment or runtime schema owned by user OWBDPL. The user needs to ensure this default location points to the current database by checking its properties.

Note: It is important to deploy the FEM_DEF_ATTR_VERSION_CODE function prior to deploying any Mapping.

Step III: Populate Staging table:

Data to be imported into Enterprise Performance Foundation should be populated in the Staging table (using SQL scripts, SQL*Loader or Oracle Warehouse Builder Mappings).

Note: It is the responsibility of the user to clear the data in the Staging table once the data has been successfully loaded into the Interface tables.

Step IV: Execute Mappings:

Once the Staging tables and Mappings are deployed successfully to the target database the Mappings are ready to be executed so that the data is moved from the Staging table to the appropriate Enterprise Performance Foundation Interface tables.

Step V: Start Dimension Loader:

Once the data has been moved from the Staging table to the Enterprise Performance Foundation Interface tables the Dimension/Fact/Hierarchy Loader programs should be run so that the data is moved the Enterprise Performance Foundation core tables.

Importing Fact/Hierarchy data into Enterprise Performance Foundation

Importing Fact/Hierarchy data into Enterprise Performance Foundation requires the following steps:

Step I: Create Mapping:

Mapping should be created between your source of the Fact/Hierarchy data and the Enterprise Performance Foundation Fact/Hierarchy interface tables. Please refer to *Oracle Warehouse Builder 10gR2 Users Guide* for more information.

Step II: Deploy Mapping:

Mapping should be deployed to the default location called OWBDPL_LOCATION successfully.

Step III: Execute Mapping:

Execute the Mapping so that data is moved from your source to the Enterprise Performance Foundation Interface tables.

Step IV: Start Fact/Hierarchy Loader:

Once the Fact/Hierarchy interface tables are populated start the Fact/Hierarchy Loader program to move the data to the Enterprise Performance Foundation core tables.

Index

A

- administration
 - dimensions, 10-6
 - table and column registration, 10-1
- administrative tasks, 10-1
- application preferences, 1-4
- attributes
 - about, 1-2
 - working with, 3-2

B

- business events, D-1

C

- classifications for tables, 10-4
- column mappings for interface tables, 10-4
- column registration, 10-1
- columns
 - display names, specifying, 10-6
- concurrent programs
 - about, 8-1
 - descriptions, 8-3
- conditions
 - about, 4-1
 - creating, 4-4
 - data components, 4-7
 - dimension components, 4-4
 - working with, 4-1

D

- data components for conditions, 4-7
- data inspector
 - about, 5-1
- data inspector rules
 - about, 5-1
 - creating, 5-2
 - deleting, 5-5
 - duplicating, 5-4
 - running, 5-3
 - updating rule names and descriptions, 5-4
 - updating selection settings, 5-4
 - viewing, 5-3
- data loader programs

- Detail Client Data Loader, A-88
- Dimension Hierarchy Loader, A-52
- Dimension Member Loader, A-7
- External General Ledger Data Loader, A-72
- data organization, 1-1
- data set groups
 - about, 7-1
 - creating, 7-2
 - deleting, 7-5
 - duplicating, 7-3
 - updating, 7-4
 - viewing, 7-3
- data sets, 7-1
- database links
 - See* registered database links
- DataX Loader program, A-1
- Detail Client Data Loader program, 8-4, A-88
- dimension attributes
 - working with, 3-2
- dimension components for conditions, 4-4
- dimension defaults, 3-22
- dimension definitions
 - working with, 3-1
- dimension hierarchies
 - working with, 3-11
- Dimension Hierarchy Loader program, 8-4, A-52
- Dimension Hierarchy Migration program, 8-5
- dimension levels
 - working with, 3-5
- Dimension Member Loader program, 8-5, A-7
- Dimension Member Migration program, 8-6
- dimension members
 - about, 1-1
 - working with, 3-7
- dimension properties
 - specifying, 10-7
- dimensions
 - about, 1-1
 - administration tasks for, 10-6
 - dimension properties, specifying, 10-7
 - display names, specifying, 10-6
- Discoverer End business area
 - See* Oracle Discoverer business area
- Discoverer Hierarchy Transformation Engine program, 8-8
- display names

for columns, 10-6
for dimensions, 10-6

E

Enterprise Performance Foundation
description, 1-1
EPF - Core Business Area, C-1
Executed Rule Removal program, 8-6
Executed Rule Request Removal program, 8-7
External General Ledger Data Loader program,
8-7, A-72

F

fact tables, 1-2
folders
See security folders

G

General Ledger Balances Rule Engine program,
8-8
General Ledger Calendar Rule Engine program,
8-8
General Ledger Dimension Rule Engine program,
8-8
General Ledger Hierarchy Rule Engine program,
8-9
global value set combinations
relationship to value sets and ledgers, 1-2
working with, 10-8

H

hierarchies
about, 1-2
working with, 3-11
Home page
customizing, 10-15
description, 1-3

I

interface table column mappings, 10-4
interface tables and external data loaders, A-1

L

ledgers
relationship to value sets and global value set
combinations, 1-2
levels
about, 1-2
working with, 3-5

M

members

working with, 3-7
metadata organization, 1-1

O

Oracle Approvals Management (AME), 11-1
Oracle Discoverer business area, C-1
Oracle Workflow, 11-1

P

process locks
about, 9-1
removing, 9-2
process tuning options
working with, 10-10
processing keys for tables, 10-4
profile options
Application Preferences profile options, B-3
Enterprise Performance Foundation profile
options, B-1
Program - Enterprise Performance Foundation
Discoverer Hierarchy Transformation Engine
program, 8-8
Program - Enterprise Performance Foundation
General Ledger Balances Rule Engine program,
8-8
Program - Enterprise Performance Foundation
General Ledger Calendar Rule Engine program,
8-8
Program - Enterprise Performance Foundation
General Ledger Dimension Rule Engine program,
8-8
Program - Enterprise Performance Foundation
General Ledger Hierarchy Rule Engine program,
8-9
programs
concurrent programs, 8-1

R

registered database links
working with, 10-9
requests
about, 8-1
submitting, 8-1
viewing, 8-9
responsibilities, user, 1-4
rule execution results
about, 9-1
removing, 9-2
viewing, 9-2
rule sets
about, 6-1
creating, 6-1
deleting, 6-4
duplicating, 6-3
running, 6-2

- updating definition settings, 6-3
- updating names and descriptions, 6-3
- viewing, 6-2
- rules
 - approval process for, 11-2
 - approval process for, as related to production data sets, 11-1
 - deletion process for, 11-2
- rules, data inspector
 - See* data inspector rules

S

- security folders
 - about, 10-13
 - creating, 10-14
 - deleting, 10-15
 - duplicating, 10-15
 - updating, 10-15
 - viewing, 10-14
- Sign Extended Account Types program, 8-7

T

- table registration, 10-1
- tables
 - classifications, 10-4
 - column display names, 10-3
 - interface table column mappings, 10-4
 - processing keys, 10-4
 - property assignments for columns, 10-5
 - registering, 10-2
- tuning options
 - working with, 10-10

U

- user responsibilities, 1-4

V

- value sets
 - relationship to global value set combinations and ledgers, 1-2
 - working with, 10-7

