

# **Oracle® Fusion Accounting Hub Implementation Guide**

11g Release 6 (11.1.6)

**Part Number E20374-06**

September 2012

Oracle® Fusion Accounting Hub Implementation Guide

Part Number E20374-06

Copyright © 2011-2012, Oracle and/or its affiliates. All rights reserved.

Authors: Kathryn Wahnoutka, Asra Alim, Tina Brand, Essan Ni Jirman, P. S. G. V. Sekhar, Barbara Snyder, Megan Wallace

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

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

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

---

---

# Contents

## 1 Fusion Accounting Hub Overview

Oracle Fusion Accounting Hub Features: Overview .....	1-1
Oracle Fusion Accounting Hub: How It Works .....	1-2
Accounting Configuration Offerings: Overview .....	1-5
Implementation Options for Oracle Fusion Accounting Hub: Overview .....	1-6
Manage Application Implementation .....	1-8

## 2 Define Enterprise Structures for Fusion Accounting Hub

Enterprise Structures: Overview .....	2-1
Enterprise Structures Business Process Model: Explained .....	2-3
Global Enterprise Configuration: Points to Consider .....	2-5
Modeling Your Enterprise Management Structure in Oracle Fusion: Example .....	2-5
Define Common Financials Configuration for Rapid Implementation .....	2-8
Define Initial Configuration with the Enterprise Structures Configurator .....	2-18
Define Enterprise for Fusion Accounting Hub .....	2-51

## 3 Define Financial Reporting Structures

Representing Your Enterprise Structure in Your Financial Reporting Structure: Overview .....	3-1
Financial Enterprise Structure: How It Fits Together .....	3-1
Modeling Your Financial Reporting Structure in Oracle Fusion: Example .....	3-4
Define Chart of Accounts .....	3-7
Manage Value Sets .....	3-20
Manage Account Hierarchies .....	3-24
Importing Segment Values and Hierarchies: Explained .....	3-39
Maintain Segment Value Attributes .....	3-42
Deploy Flexfields .....	3-50
Manage Cross Validation Rules .....	3-58
Creating a Cross Validation Rule: Example .....	3-59
Manage Chart of Accounts Mapping .....	3-60
Manage DRM Synchronization .....	3-62
Define Calendars .....	3-62
Define Currencies .....	3-66
Manage Conversion Rate Types .....	3-68
Manage Daily Rates .....	3-72

## 4 Define Ledgers

Ledgers and Subledgers: Explained .....	4-1
---	-----

Ledgers: Points to Consider .....	4-2
Financial Ledgers: How They Fit Together .....	4-5
Creating Primary Ledgers: Example .....	4-8
Specifying Ledger Options: Worked Example .....	4-8
Assigning Legal Entities and Balancing Segments: Examples .....	4-11
Review and Submit Accounting Configuration .....	4-12
Manage Reporting Currencies .....	4-31
Manage Journal Approval Rules .....	4-34
Manage AutoPost Criteria Sets .....	4-36
Manage Journal Reversal Criteria Sets .....	4-39

## 5 Define Period Close Components

Manage Period Close .....	5-1
Period Close Components: Explained .....	5-4
Manage Allocations and Periodic Entries .....	5-5
Manage Revaluations .....	5-16

## 6 Define Applications Coexistence Configuration

Integration with Oracle E-Business Suite and Oracle PeopleSoft: Overview .....	6-1
--	-----

## 7 Define Accounting Transformation

Accounting Transformation Configuration: Overview .....	7-1
Analyze Accounting Events .....	7-11
Register Source Systems .....	7-19
Implement Accounting Event Interfaces .....	7-34
Create Accounting Integration .....	7-54
Manage Accounting Rules .....	7-59
Migrate the Configuration .....	7-74
Secure Accounting Transformations .....	7-75
Create and Process Subledger Journal Entries .....	7-76

## 8 Manage Subledger Accounting

Manage Subledger Accounting: Overview .....	8-1
Create Accounting .....	8-1
Create Subledger Journal Adjustment .....	8-5
Review Subledger Journal Entry .....	8-9
Subledger Accounting Reports: Overview .....	8-10

## 9 Define and Maintain Intercompany Processing Rules

Manage Intercompany System Options .....	9-1
Manage Intercompany Organizations .....	9-3
Define Invoicing Options .....	9-4

Manage Intercompany Balancing Rules .....	9-5
Manage Ledger Balancing Options .....	9-13
Manage Intercompany Transactions .....	9-18
Reconcile Accounts .....	9-21

## 10 Define Hyperion Financial Management Integration

Oracle Hyperion Financial Management: Overview .....	10-1
Integration with Hyperion Financial Management: Overview .....	10-2

## 11 Define Budget Configuration

Budget Uploads: Overview .....	11-1
Importing Budget Data from a Flat File: Explained .....	11-2
Loading Data to the Budget Interface Table: Explained .....	11-3
Importing Budget Data from a Spreadsheet: Explained .....	11-4
Budget Correction with Oracle ADF Desktop Integrator: Explained .....	11-6
Creating Budget Data Security: Worked Example .....	11-6
Oracle Hyperion Planning: Overview .....	11-8
Integration with Hyperion Planning: Overview .....	11-9

## 12 Define Financial Reporting

Financial Reporting Center: How It Works .....	12-1
Setting Up Your Financial Reporting Center: Critical Choices .....	12-2
Oracle Fusion General Ledger Predefined Reports .....	12-6

## 13 Define Extensions for Fusion Accounting Hub

Define Custom Enterprise Scheduler Jobs for Fusion Accounting Hub .....	13-1
---	------



---

---

# Preface

This Preface introduces the guides, online help, and other information sources available to help you more effectively use Oracle Fusion Applications.

## Oracle Fusion Applications Help

You can access Oracle Fusion Applications Help for the current page, section, activity, or task by clicking the help icon. The following figure depicts the help icon.



You can add custom help files to replace or supplement the provided content. Each release update includes new help content to ensure you have access to the latest information. Patching does not affect your custom help content.

## Oracle Fusion Applications Guides

Oracle Fusion Applications guides are a structured collection of the help topics, examples, and FAQs from the help system packaged for easy download and offline reference, and sequenced to facilitate learning. You can access the guides from the **Guides** menu in the global area at the top of Oracle Fusion Applications Help pages.

---

### Note

The **Guides** menu also provides access to the business process models on which Oracle Fusion Applications is based.

---

Guides are designed for specific audiences:

- **User Guides** address the tasks in one or more business processes. They are intended for users who perform these tasks, and managers looking for an overview of the business processes. They are organized by the business process activities and tasks.
- **Implementation Guides** address the tasks required to set up an offering, or selected features of an offering. They are intended for implementors. They are organized to follow the task list sequence of the offerings, as displayed within the Setup and Maintenance work area provided by Oracle Fusion Functional Setup Manager.
- **Concept Guides** explain the key concepts and decisions for a specific area of functionality. They are intended for decision makers, such as chief financial officers, financial analysts, and implementation consultants. They are organized by the logical flow of features and functions.

- **Security Reference Manuals** describe the predefined data that is included in the security reference implementation for one offering. They are intended for implementors, security administrators, and auditors. They are organized by role.

These guides cover specific business processes and offerings. Common areas are addressed in the guides listed in the following table.

Guide	Intended Audience	Purpose
Common User Guide	All users	Explains tasks performed by most users.
Common Implementation Guide	Implementors	Explains tasks within the Define Common Applications Configuration task list, which is included in all offerings.
Information Technology Management, Implement Applications Guide	Implementors	Explains how to use Oracle Fusion Functional Setup Manager to plan, manage, and track your implementation projects, migrate setup data, and validate implementations.
Technical Guides	System administrators, application developers, and technical members of implementation teams	Explain how to install, patch, administer, and customize Oracle Fusion Applications.

For guides that are not available from the Guides menu, go to Oracle Technology Network at <http://www.oracle.com/technetwork/indexes/documentation>.

## Other Information Sources

### My Oracle Support

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

Use the My Oracle Support Knowledge Browser to find documents for a product area. You can search for release-specific information, such as patches, alerts, white papers, and troubleshooting tips. Other services include health checks, guided lifecycle advice, and direct contact with industry experts through the My Oracle Support Community.

### Oracle Enterprise Repository for Oracle Fusion Applications

Oracle Enterprise Repository for Oracle Fusion Applications provides visibility into service-oriented architecture assets to help you manage the lifecycle of your software from planning through implementation, testing, production,

and changes. In Oracle Fusion Applications, you can use the Oracle Enterprise Repository for Oracle Fusion Applications at <http://fusionappsoer.oracle.com> for:

- Technical information about integrating with other applications, including services, operations, composites, events, and integration tables. The classification scheme shows the scenarios in which you use the assets, and includes diagrams, schematics, and links to other technical documentation.
- Publishing other technical information such as reusable components, policies, architecture diagrams, and topology diagrams.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/us/corporate/accessibility/index.html>.

## Comments and Suggestions

Your comments are important to us. We encourage you to send us feedback about Oracle Fusion Applications Help and guides. Please send your suggestions to [oracle\\_fusion\\_applications\\_help\\_ww\\_grp@oracle.com](mailto:oracle_fusion_applications_help_ww_grp@oracle.com). You can use the **Send Feedback to Oracle** link in the footer of Oracle Fusion Applications Help.



---

---

# Fusion Accounting Hub Overview

## Oracle Fusion Accounting Hub Features: Overview

Oracle Fusion Accounting Hub (FAH) provides a complete set of accounting tools and unparalleled access to financial data.

It includes:

- Oracle Fusion General Ledger best of breed features that provide journal entry import and creation, real time balances, accounting controls, and close functionality. The close functionality contains robust intercompany balancing, an calculation manager for the definition of allocation rules using complex formulas, automatic generation of allocation journals, enhanced journal approval, and year end process management.
- Oracle Fusion Financial Reporting that provides embedded balances cube functionality with multidimensional, online analytical processing (OLAP) of financial information. Quickly slice and dice data across dimensions and drill up, down, and across on any parent level within the chart of accounts.
- Oracle Fusion Subledger Accounting rules that provide flexible transformation of transaction and reference information from diverse, non-Oracle, industry applications into accurate, detailed, auditable accounting. Subledger Accounting is also embedded in the Oracle Fusion subledgers such as Payables, Receivables, Assets, and Inventory.
- Oracle Hyperion Data Relationship Management, Fusion Edition integration that provides the opportunity to perform charts of accounts and hierarchies maintenance. Data Relationship Management is a master data management application and requires licensing. The functionality also establishes corporate wide accounting structures, for Oracle and non-Oracle ledgers, and automatically updates charts of accounts and hierarchies across multiple ledgers.
- Applications coexistence integration with the E-Business Suite (EBS) and PeopleSoft General Ledgers that provides the opportunity to leverage the Oracle Fusion Accounting Hub with minimal implementation effort. Continue to use your E-Business Suite or PeopleSoft applications for transaction processing, while taking advantage of the Oracle Fusion Accounting Hub for financial reporting and analysis.
- Oracle Hyperion Planning, Fusion Edition and Oracle Hyperion Financial Management, Fusion Edition integration that provides the opportunity

to perform your planning, budgeting, and consolidation functions in an integrated accounting environment. Both products required licensing.

## Oracle Fusion Accounting Hub: How It Works

The Oracle Fusion Accounting Hub process can begin by using financial data from any or all of the following:

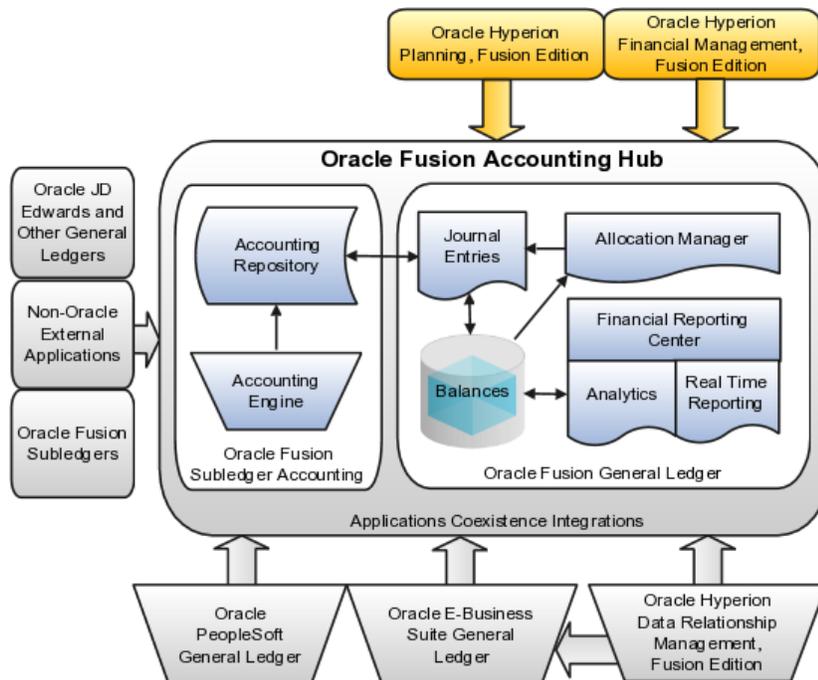
- Non-Oracle external applications including transaction and reference information from industry, specific applications
- Oracle Fusion subledgers including subledger journals
- Oracle E-Business Suite, Oracle PeopleSoft, and Oracle JD Edwards General Ledgers including account balances

The Accounting Hub process ends with complete reporting and analysis solutions.

## The Oracle Fusion Accounting Hub Components

The Accounting Hub contains the following components:

- Oracle Fusion Subledger Accounting to perform accounting transformations on external system data
- Oracle Fusion General Ledger
- Oracle Fusion Financial Reporting Center
- Integration with Oracle Hyperion Data Management, Fusion Edition for chart of accounts and hierarchy maintenance
- Applications coexistence integration with the Oracle E-Business Suite and Oracle PeopleSoft General Ledgers
- Integration with Oracle Hyperion Planning, Fusion Edition and Oracle Hyperion Financial Management, Fusion Edition



## Oracle Fusion Subledger Accounting for External Systems

The Oracle Fusion Accounting Hub allows you to use Subledger Accounting to perform accounting transformations on external, non-Oracle system data. Subledger Accounting is also used to flexibly create accounting for Oracle subledgers such as Oracle Payables and Receivables. Subledger Accounting includes:

- Registration of your external systems, indicating what types of transactions or activities require accounting from those systems.
- Create a library of transaction and reference information that will be used for defining accounting treatments.
- Configurable accounting rules to define accounting treatments for transactions.
- Accounting engine that combines transaction and reference information from source systems with accounting rules to create detailed journals stored in an accounting repository.
- Detailed subledger accounting journal entry repository to audit and reconcile accounting balances.
- Reports and user interface inquiries for analyzing accounting transformations.
- Configurable analytical balances based upon source system or reference attributes.

## Oracle Fusion General Ledger

The Oracle Fusion General Ledger combines traditional general ledger functionality with embedded Oracle Hyperion Essbase functionality. The General Ledger functionality includes:

- Journal entry creation, including updating of account balances cubes and tables.
- Date effective trees for chart of accounts maintenance and financial reporting, including what if analysis.
- Automatic balances cube creation from the chart of accounts configuration, simplifying implementation.
- Preaggregated balances at every summarization level across each dimension of the chart of accounts and accounting periods, improving reporting performance.
- Multidimensional analysis using dimensions, such as the chart of accounts, periods, and currency, to provide drill down and drill through functionally.
- Drill to details from summary balances.
- Intelligence and analytics embedded within the context of a journal entry, enabling quick and accurate completion of the journal entry process. As journal entries are entered, a what if analysis to determine the impact

of the unposted journals on account balances is displayed in the user interface, eliminating the need to navigate to an inquiry page or run a report to verify the results.

- Three balancing segments available in your chart of accounts, enabling more detailed reporting.
- Automatic intercompany balancing journal creation in both Subledger Accounting and General Ledger applications, ensuring proper recording of transactions across legal entities.
- Calculation Manager rules using complex formulas to distribute revenue and costs throughout the organization, enabling consistent periodic generation of allocation journal entries.

### **Financial Reporting Center**

The Oracle Fusion General Ledger provides a Financial Reporting Center with robust financial reporting and analysis using data from your balances cubes. The dimensions contained in your chart of account segments become the direct source of multidimensional analysis. Direct links are maintained to your transactional data permitting comprehensive drill down from journals to transaction details. Use the following tools for your reporting and analysis:

- Financial Reporting to generate your reports
- Smart View to generate spreadsheet reports
- Oracle Transaction Business Intelligence to report using embedded analytics
- Oracle Business Intelligence Analytics to provide real time, ad hoc queries from a data warehouse
- General Accounting and Journals dashboards to perform online inquiry and dashboard publication
- Account Monitor and Account Inspector to perform online multidimensional analysis of accounting balances

The Oracle Fusion Accounting Hub fits into Oracle's broader enterprise performance management framework via integration with Oracle Hyperion Planning, Fusion Edition and Oracle Hyperion Financial Management, Fusion Edition, You can perform:

- Financial consolidations using Financial Management
- Planning and budgeting using Planning

### **Integration with Hyperion Data Management**

Oracle Fusion Accounting Hub is integrated with Oracle Hyperion Data Relationship Management, Fusion Edition, which is a master data management solution for creating and maintaining hierarchies across your enterprise. This integration allows you to maintain your charts of accounts values and hierarchies in one central location, and then to synchronize your hierarchies in Oracle Fusion and E-Business Suite General Ledgers. With licensing and integration of Data Relationship Management, you can:

- Update and change Data Relationship Management versions and hierarchies. The results of these changes are exported to the values sets in the Oracle Fusion and E-Business Suite General Ledgers, E-Business Suite parent child hierarchies, and Oracle Fusion Trees.
- Synchronize charts of accounts and hierarchies across multiple Oracle general ledger instances.
- Store segment value attributes, such as account type, start date, and end date, providing the ability to enter such values in Data Relationship Management and integrate them to Oracle Fusion and E-Business Suite General Ledgers.
- Submit standardized E-Business Suite Standard Request Submission and Oracle Fusion Enterprise Schedule Service requests from both the E-Business Suite and the Oracle Fusion General Ledgers to import values from Data Relationship Management.

### **Integration with E-Business Suite and PeopleSoft General Ledgers**

Integration between the Oracle E-Business Suite, Oracle PeopleSoft, and Oracle Fusion General Ledgers can be used to support an applications coexistence strategy. This strategy permits you to continue to use your E-Business Suite or PeopleSoft applications while also using the Oracle Fusion Accounting Hub. Application coexistence provides the ability to:

- Expand your reporting and analytical capabilities using the Financial Reporting Center and dashboard features of the Oracle Fusion Accounting Hub. This functionality includes the ability to drill down to E-Business Suite or PeopleSoft account balances and transactions reflected in the Oracle Fusion reports and dashboards.
- Continue to use your E-Business Suite or PeopleSoft applications for your procurement, payables, receivables, and any other E-Business Suite or PeopleSoft subledger processes without disruptions.

## **Accounting Configuration Offerings: Overview**

The Setup and Maintenance work area in the Oracle Fusion Applications is used to manage the configuration of legal entities, ledgers, and reporting currencies that comprise your accounting configuration. To create a new legal entity or ledger, your implementation consultant or system administrator must create an implementation project. This implementation project can be populated by either adding a financials related offering or one or more task lists.

---

### **Note**

Setup tasks that are not related to the ledger or legal entity specific setup tasks can be invoked from either an implementation project or launched directly from the Setup and Maintenance work area.

---

There are two offerings predefined for financial implementations.

- The Oracle Fusion Accounting Hub offering is used to add the Oracle Fusion General Ledger and Oracle Fusion Subledger Accounting

application features to an existing enterprise resource planning (ERP) system to enhance the current reporting and analysis.

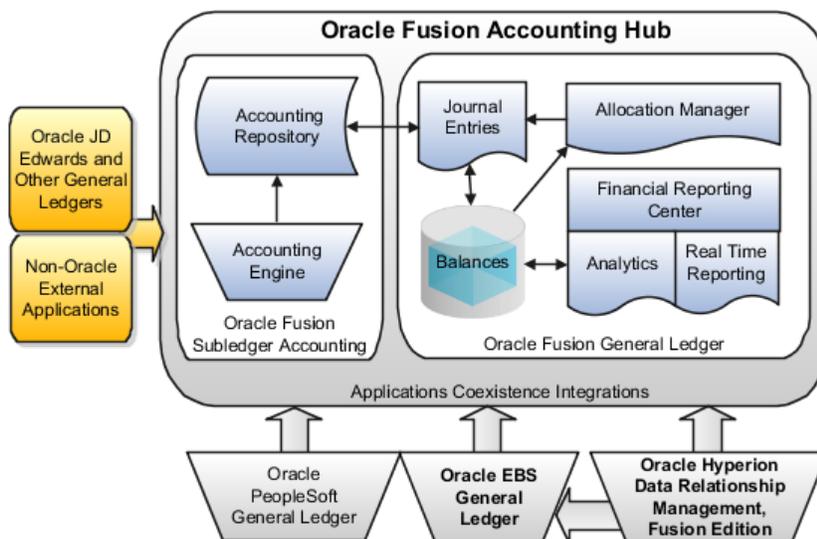
- The Oracle Fusion Financials offering, which includes the Oracle Fusion General Ledger and Oracle Fusion Subledger Accounting application features, as well as at least one of the subledger financial applications.

When adding an offering to an implementation project, implementation consultants can customize the tasks displayed by adding additional tasks to the implementation project.

## Implementation Options for Oracle Fusion Accounting Hub: Overview

The Oracle Fusion Accounting Hub has three possible implementation scenarios that can be used separately or in conjunction with each other. These three scenarios are:

- Coexistence with Oracle E-Business Suite and other general ledgers providing enhanced allocations, reporting, and analytical functionality
- Accounting for diverse business system events and transactions with configurable and auditable rule based accounting transformations
- Expansion of financial reporting enabling creation of multidimensional, self service reporting and analytics with real-time accounting information from a single source with flexible formatting options.



### Coexistence with E-Business Suite and PeopleSoft

Oracle Fusion Accounting Hub provides embedded integration with the Oracle E-Business Suite and Oracle PeopleSoft General Ledgers. The integration has three main components:

- Transfer of accounting balances from the E-Business Suite and PeopleSoft General Ledgers to the Oracle Fusion Accounting Hub General Ledger

- Drill down from the Oracle Fusion Accounting Hub General Ledger to the E-Business Suite and PeopleSoft General Ledgers
- Synchronization of charts of accounts segment values and hierarchies using Oracle Hyperion Data Relationship Management, Fusion Edition, with E-Business Suite and Oracle Fusion General Ledger. Licensing of Data Relationship Management is required.

See the following guides for details on implementing this scenario:

- Oracle General Ledger Implementation Guide Release 12.2 for Oracle E-Business Suite General Ledger
- Oracle Hyperion Data Relationship Management, Fusion Edition Administrator's Guide for Oracle Hyperion Data Relationship Management
- Oracle Hyperion Data Relationship Management, Fusion Edition Users Guide for Oracle Hyperion Data Relationship Management

### **Accounting for Diverse Transactions**

The Oracle Fusion Accounting Hub provides accounting for diverse business system events and transactions using configurable and auditable rule based accounting transformations. The Accounting Hub process begins with transactions and data from your non-Oracle, industry specific subledgers and other general ledgers. Subledger and general ledger applications that generate accounting events invoke the Create Accounting process to create journals that can be posted to the Oracle Fusion General Ledger. The process ends with complete reporting and analysis solutions.

See the following for details on implementing this scenario: Define Accounting Transformation Configuration chapter in the Oracle Fusion Accounting Hub Implementation Guide.

### **Expansion of Financial Reporting**

Oracle Fusion Accounting Hub enables creation of multidimensional, self service reporting and analytics with real time accounting information from a single source with flexible formatting options. The Financial Reporting Center includes predefined financial reports, the ability to create new reports, and flexible options for dissemination of such reports. The Financial Reporting Center also includes the Account Monitor and Inspector, providing drill down and pivot views of your balances. Oracle Hyperion Smart View provides spreadsheet analytics and Oracle Transaction Business Intelligence, Oracle Business Intelligence Analytics, and Oracle Business Intelligence Publisher provide key performance indicators (KPI), dashboards, and flexible reporting.

See the following guides for details on implementing this scenario:

- Oracle Fusion Accounting Hub Implementation Guide for the Oracle Fusion Accounting Hub
- Oracle Essbase Database Administrator's Guide for Oracle Essbase
- Oracle Hyperion Financial Reporting Administrator's Guide for Oracle Hyperion Financial Reporting

- Oracle Enterprise Performance Management Workspace Administrator's Guide for Oracle Enterprise Performance Management
- Oracle Hyperion Smart View for Office User's Guide for Oracle Hyperion Smart View
- Oracle Hyperion Financial Reporting Studio User's Guide for Oracle Hyperion Financial Reporting Studio

## Manage Application Implementation

### Manage Application Implementation: Overview

The Manage Applications Implementation business process enables rapid and efficient planning, configuration, implementation, deployment, and ongoing maintenance of Oracle Fusion applications through self-service administration.

The Setup and Maintenance work area offers you the following benefits:

- Prepackaged lists of implementation tasks  
 Task lists can be easily configured and extended to better fit with business requirements. Auto-generated, sequential task lists include prerequisites and address dependencies to give full visibility to end-to-end setup requirements of Oracle Fusion applications.
- Rapid start  
 Specific implementations can become templates to facilitate reuse and rapid-start of consistent Oracle Fusion applications setup across many instances.
- Comprehensive reporting  
 A set of built-in reports helps to analyze, validate and audit configurations, implementations, and setup data of Oracle Fusion applications.

With Oracle Fusion Functional Setup Manager you can:

- Learn about and analyze implementation requirements.
- Configure Oracle Fusion applications to match your business needs.
- Get complete visibility to setup requirements through guided, sequential task lists downloadable into Excel for project planning.
- Enter setup data through easy-to-use user interfaces available directly from the task lists.
- Export and import data from one instance to another for rapid setup.
- Validate setup by reviewing setup data reports.
- Implement all Oracle Fusion applications through a standard and consistent process.

There are several documentation resources available for learning how to configure Oracle Fusion Applications.

- Functional Setup Manager Developer's Guide
- Common Implementation Guide
- Customer Data Management Implementation Guide
- Enterprise Contracts Implementation Guide
- Marketing Implementation Guide
- Sales Implementation Guide
- Fusion Accounting Hub Implementation Guide
- Financials Implementation Guide
- Compensation Management Implementation Guide
- Workforce Deployment Implementation Guide
- Workforce Development Implementation Guide
- Incentive Compensation Implementation Guide
- Procurement Implementation Guide
- P6 EPPM Administrator's Guide for an Oracle Database
- P6 EPPM Administrator's Guide for Microsoft SQL Server Database

## Implementation Projects: Explained

An implementation project is the list of setup tasks you need to complete to implement selected offerings and options. You create a project by selecting the offerings and options you want to implement together. You manage the project as a unit throughout the implementation lifecycle. You can assign these tasks to users and track their completion using the included project management tools.

### Maintaining Setup Data

You can also create an implementation project to maintain the setup of specific business processes and activities. In this case, you select specific setup task lists and tasks

### Exporting and Importing

Implementation projects are also the foundation for setup export and import. You use them to identify which business objects, and consequently setup data, you will export or import and in which order.

### Selecting Offerings

When creating an implementation project you see the list of offerings and options that are configured for implementation. Implementation managers

specify which of those offerings and options to include in an implementation project. There are no hard and fast rules for how many offerings you should include in one implementation project. The implementation manager should decide based on how they plan to manage their implementations. For example, if you will implement and deploy different offerings at different times, then having separate implementation projects will make it easier to manage the implementation life cycles. Furthermore, the more offerings you included in an implementation project, the bigger the generated task list will be. This is because the implementation task list includes all setup tasks needed to implement all included offerings. Alternatively, segmenting into multiple implementation projects makes the process easier to manage.

## Offerings: Explained

Offerings are application solution sets representing one or more business processes and activities that you typically provision and implement as a unit. They are, therefore, the primary drivers of functional setup of Oracle Fusion applications. Some of the examples of offerings are Financials, Procurement, Sales, Marketing, Order Orchestration, and Workforce Deployment. An offering may have one or more options or feature choices.

### Implementation Task Lists

The configuration of the offerings will determine how the list of setup tasks is generated during the implementation phase. Only the setup tasks needed to implement the selected offerings, options and features will be included in the task list, giving you a targeted, clutter-free task list necessary to meet your implementation requirements.

### Enabling Offerings

Offerings and their options are presented in an expandable and collapsible hierarchy to facilitate progressive decision making when specifying whether or not an enterprise plans to implement them. An offering or its options can either be selected or not be selected for implementation. Implementation managers decide which offerings to enable.

### Provisioning Offerings

The Provisioned column on the Configure Offerings page shows whether or not an offering is provisioned. While you are not prevented from configuring offerings that have not been provisioned, ultimately the users are not able to perform the tasks needed to enter setup data for those offerings until appropriate enterprise applications (Java EE applications) are provisioned and their location (end point URLs) is registered.

## Options: Explained

Each offering in general includes a set of standard functionality and a set of optional modules, which are called options. For example, in addition to standard Opportunity Management, the Sales offering includes optional functionality

such as Sales Catalog, Sales Forecasting, Sales Prediction Engine, and Outlook Integration. These optional functions may not be relevant to all application implementations. Because these are subprocesses within an offering, you do not always implement options that are not core to the standard transactions of the offering.

## Feature Choices: Explained

Offerings include optional or alternative business rules or processes called feature choices. You make feature selections according to your business requirements to get the best fit with the offering. If the selected offerings and options have dependent features then those features are applicable when you implement the corresponding offering or option. In general, the features are set with a default configuration based on their typical usage in most implementations. However, you should always review the available feature choices for their selected offerings and options and configure them as appropriate for the implementation.

You can configure feature choices in three different ways:

### Yes or No

If a feature can either be applicable or not be applicable to an implementation, a single checkbox is presented for selection. Check or uncheck to specify yes or no respectively.

### Single Select

If a feature has multiple choices but only one can be applicable to an implementation, multiple choices are presented as radio buttons. You can turn on only one of those choices.

### Multi-Select

If the feature has multiple choices but one or more can be applicable to an implementation then all choices are presented with a checkbox. Select all that apply by checking the appropriate choices.



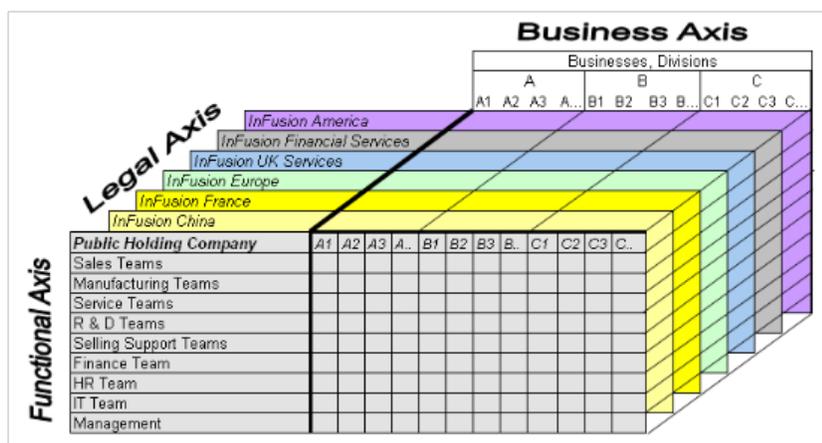
## Define Enterprise Structures for Fusion Accounting Hub

### Enterprise Structures: Overview

Oracle Fusion Applications have been designed to ensure your enterprise can be modeled to meet legal and management objectives. The decisions about your implementation of Oracle Fusion Applications are affected by your:

- Industry
- Business unit requirements for autonomy
- Business and accounting policies
- Business functions performed by business units and optionally, centralized in shared service centers
- Locations of facilities

Every enterprise has three fundamental structures, legal, managerial, and functional, that are used to describe its operations and provide a basis for reporting. In Oracle Fusion, these structures are implemented using the chart of accounts and organizations. Although many alternative hierarchies can be implemented and used for reporting, you are likely to have one primary structure that organizes your business into divisions, business units, and departments aligned by your strategic objectives.



## Legal Structure

The figure above shows a typical group of legal entities, operating various business and functional organizations. Your ability to buy and sell, own, and employ comes from your charter in the legal system. A corporation is a distinct legal entity from its owners and managers. The corporation is owned by its shareholders, who may be individuals or other corporations. There are many other kinds of legal entities, such as sole proprietorships, partnerships, and government agencies.

A legally recognized entity can own and trade assets and employ people in the jurisdiction in which it is registered. When granted these privileges, legal entities are also assigned responsibilities to:

- Account for themselves to the public through statutory and external reporting
- Comply with legislation and regulations
- Pay income and transaction taxes
- Process value added tax (VAT) collection on behalf of the taxing authority

Many large enterprises isolate risk and optimize taxes by incorporating subsidiaries. They create legal entities to facilitate legal compliance, segregate operations, optimize taxes, complete contractual relationships, and isolate risk. Enterprises use legal entities to establish their enterprise's identity under the laws of each country in which their enterprise operates.

In the figure above, a separate card represents a series of registered companies. Each company, including the public holding company, InFusion America, must be registered in the countries where they do business. Each company consists of various divisions created for purposes of management reporting. These are shown as vertical columns on each card. For example, a group might have a separate company for each business in the United States (US), but have their United Kingdom (UK) legal entity represent all businesses in that country. The divisions are linked across the cards so that a business can appear on some or all of the cards. For example, the air quality monitoring systems business might be operated by the US, UK, and France companies. The list of business divisions is on the Business Axis. Each company's card is also horizontally striped by functional groups, such as the sales team and the finance team. This functional list is called the Functional Axis. The overall image suggests that information might, at a minimum, be tracked by company, business, division, and function in a group environment. In Oracle Fusion Applications, the legal structure is implemented using legal entities.

## Management Structure

Successfully managing multiple businesses requires that you segregate them by their strategic objectives, and measure their results. Although related to your legal structure, the business organizational hierarchies do not need to be reflected directly in the legal structure of the enterprise. The management structure can include divisions, subdivisions, lines of business, strategic business units, and cost centers. In the figure above, the management structure is shown on the Business Axis. In Oracle Fusion Applications, the management structure is implemented using divisions and business units.

## Functional Structure

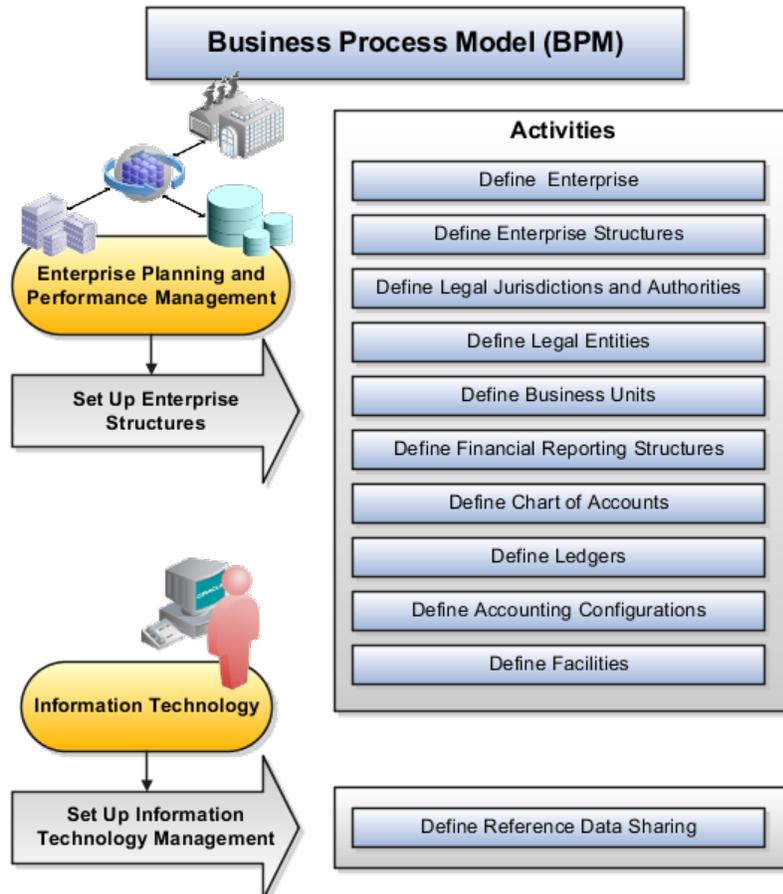
Straddling the legal and business organizations is a functional organization structured around people and their competencies. For example, sales,

manufacturing, and service teams are functional organizations. This functional structure is represented by the Functional Axis in the figure above. You reflect the efforts and expenses of your functional organizations directly on the income statement. Organizations must manage and report revenues, cost of sales, and functional expenses such as research and development (R&D) and selling, general, and administrative (SG&A) expenses. In Oracle Fusion Applications, the functional structure is implemented using departments and organizations, including sales, marketing, project, cost, and inventory organizations.

## Enterprise Structures Business Process Model: Explained

In Oracle Fusion Applications, the Enterprise Performance and Planning Business Process Model illustrates the major implementation tasks that you perform to create your enterprise structures. This process model includes the Set Up Enterprise Structures business process, which consist of implementation activities that span many product families. Information Technology is a second Business Process Model which contains the Set Up Information Technology Management business process. Define Reference Data Sharing is one of the activities in this business process and is important in the implementation of the enterprise structures. This activity creates the mechanism to share reference data sets across multiple ledgers, business units, and warehouses, reducing the administrative burden and decreasing the time needed to implement.

The following figure and chart describes the Business Process Model structures and activities.



BPM Activities	Description
Define Enterprise	Define the enterprise to capture the name of the deploying enterprise and the location of the headquarters. There is normally a single enterprise organization in a production environment. Multiple enterprises are defined when the system is used to administer multiple customer companies, or when you choose to set up additional enterprises for testing or development.
Define Enterprise Structures	Define enterprise structures to represent an organization with one or more legal entities under common control. Define internal and external organizations to represent each area of business within the enterprise.
Define Legal Jurisdictions and Authorities	Define information for governing bodies that operate within a jurisdiction.
Define Legal Entities	Define legal entities and legal reporting units for business activities handled by the Oracle Fusion Applications.
Define Business Units	Define business units of an enterprise to allow for flexible implementation, to provide a consistent entity for controlling and reporting on transactions, and to be an anchor for the sharing of sets of reference data across applications.
Define Financial Reporting Structures	Define financial reporting structures, including organization structures, charts of accounts, organizational hierarchies, calendars, currencies and rates, ledgers, and document sequences which are used in organizing the financial data of a company.
Define Chart of Accounts	Define chart of accounts including hierarchies and values to enable tracking of financial transactions and reporting at legal entity, cost center, account, and other segment levels.
Define Ledgers	Define the primary accounting ledger and any secondary ledgers that provide an alternative accounting representation of the financial data.
Define Accounting Configurations	Define the accounting configuration that serves as a framework for how financial records are maintained for an organization.
Define Facilities	Define inventory, item, and cost organizations. Inventory organizations represent facilities that manufacture or store items. The item master organization holds a single definition of items that can be shared across many inventory organizations. Cost organizations group inventory organizations within a legal entity to establish the cost accounting policies.
Define Reference Data Sharing	Define how reference data in the applications is partitioned and shared.

---

**Note**

There are product specific implementation activities that are not listed here and depend on the applications you are implementing. For example, you can

## Global Enterprise Configuration: Points to Consider

Start your global enterprise structure configuration by discussing what your organization's reporting needs are and how to represent those needs in the Oracle Fusion Applications. Consider deployment on a single instance, or at least, on as few instances as possible, to simplify reporting and consolidations for your global enterprises. The following are some questions and points to consider as you design your global enterprise structure in Oracle Fusion.

- Enterprise Configuration
- Business Unit Management
- Security Structure
- Compliance Requirements

### Enterprise Configuration

What is the level of configuration needed to achieve the reporting and accounting requirements? What components of your enterprise do you need to report on separately? Which components can be represented by building a hierarchy of values to provide reporting at both detail and summary levels? Where are you on the spectrum of centralization versus decentralization?

### Business Unit Management

What reporting do I need by business unit? How can you set up your departments or business unit accounts to achieve departmental hierarchies that report accurately on your lines of business? What reporting do you need to support the managers of your business units, and the executives who measure them? How often are business unit results aggregated? What level of reporting detail is required across business units?

### Security Structure

What level of security and access is allowed? Are business unit managers and the people that report to them secured to transactions within their own business unit? Are the transactions for their business unit largely performed by a corporate department or shared service center?

### Compliance Requirements

How do you comply with your corporate external reporting requirements and local statutory reporting requirements? Do you tend to prefer a corporate first or an autonomous local approach? Where are you on a spectrum of centralization, very centralized or decentralized?

## Modeling Your Enterprise Management Structure in Oracle Fusion: Example

This example uses a fictitious global company to demonstrate the analysis that can occur during the enterprise structure configuration planning process.

### Scenario

Your company, InFusion Corporation, is a multinational conglomerate that operates in the United States (US) and the United Kingdom (UK). InFusion

has purchased an Oracle Fusion enterprise resource planning (ERP) solution including Oracle Fusion General Ledger and all of the Oracle Fusion subledgers. You are chairing a committee to discuss creation of a model for your global enterprise structure including both your US and UK operations.

### **InFusion Corporation**

InFusion Corporation has 400 plus employees and revenue of \$120 million. Your product line includes all the components to build and maintain air quality monitoring (AQM) systems for homes and businesses. You have two distribution centers and three warehouses that share a common item master in the US and UK. Your financial services organization provides funding to your customers for the start up costs of these systems.

### **Analysis**

The following are elements you need to consider in creating your model for your global enterprise structure.

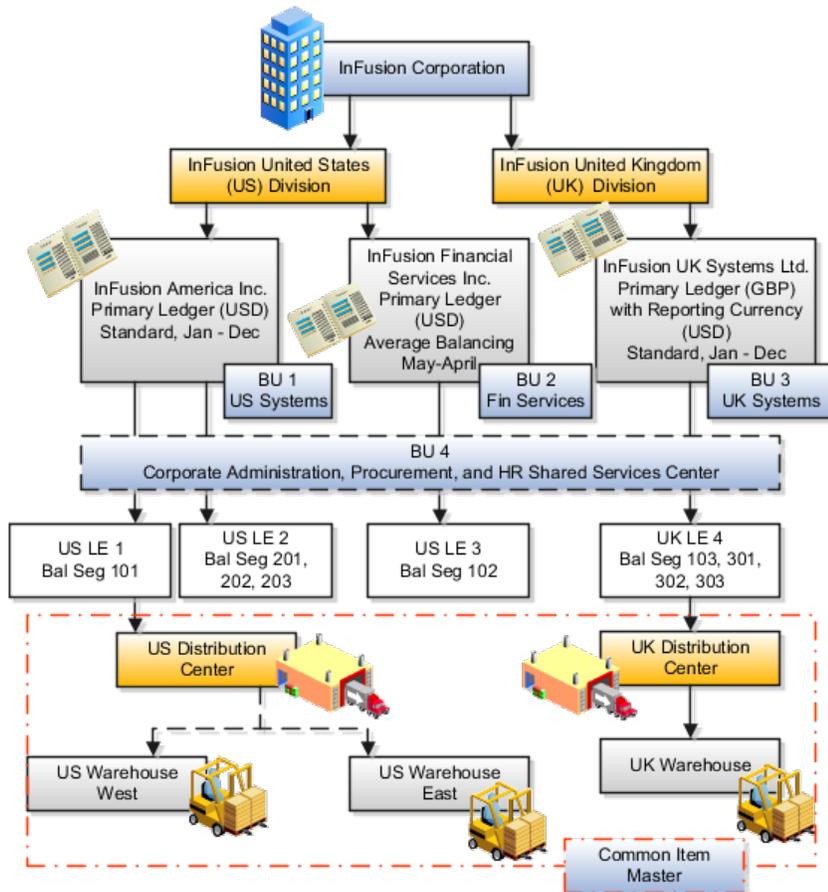
- Your company is required to report using US Generally Accepted Accounting Principles (GAAP) standards and UK Statements of Standard Accounting Practice and Financial Reporting Standards. How many ledgers do you need to achieve proper statutory reporting?
- Your managers need reports that show profit and loss (revenue and expenses) for their lines of business. Do you use business units and balancing segments to represent your divisions and businesses? Do you secure data by two segments in your chart of accounts which represents each department and legal entity or one segment that represents both to produce useful, but confidential management reports?
- Your corporate management requires reports showing total organizational performance with drill down capability to the supporting details. Do you need multiple balancing segment hierarchies to achieve proper rollup of balances for reporting requirements?
- Your company has all administrative, account payables, procurement, and human resources functions performed at their corporate headquarters. Do you need one or more business unit in which to perform all these functions? How will your shared service center be configured?

## **Global Enterprise Structure Model**

The following figure and table summarize the model that your committee has designed and uses numerical values to provide a sample representation of your structure. The model includes the following recommendations:

- Creation of three separate ledgers representing your separate legal entities:
  - InFusion America Inc.
  - InFusion Financial Services Inc.
  - InFusion UK Services Ltd.
- Consolidation of results for system components, installations, and maintenance product lines across the enterprise

- All UK general and administrative costs processed at the UK headquarters
- US Systems' general and administrative costs processed at US Corporate headquarters
- US Financial Services maintains its own payables and receivables departments



Real World Entity	Entity Name	Enterprise	Legal Entity	BSV	Ledger	BU	Cost Center	Dept	Item Master Org	Inventory Org
Enterprise	InFusion Group	M								
Company	US LE 1		M	O	O					
Company	US LE 2		M	O	O					
Company	US LE 3		M	O	O					
Company	UK LE 4		M	O	O					
Business Unit	US Systems BU 1					M				
Business Unit	FIN Services BU 2					M				
Business Unit	UK Systems BU 3					M				
Division	InFusion UK			O			O			
Division	InFusion US			O			O			
Headquarters	BU 4					M				
Shared Service Center	BU 4					M				
Department	AP Department						O	M		
List of Items	Common Item Master								O	
Distribution Center	US Distribution Center									O
Distribution Center	UK Distribution Center									O
Warehouse	US Warehouse West									O
Warehouse	US Warehouse East									O
Warehouse	UK Warehouse									O
BSV = Balancing Segment Value										
BU = Business Unit										
Dept = Department										
Org = Organization										
M = Mandatory Setup										
O = Optional Setup										

In this chart, the green globe stands for mandatory and gold globe stands for optional setup. The following statements expand on the data in the chart.

- The enterprise is mandatory because it serves as an umbrella for the entire implementation. All organizations are created within an enterprise.
- Legal entities are also mandatory. They can be optionally mapped to balancing segment values or represented by ledgers. Mapping balancing segment values to legal entities is mandatory if you plan to use the intercompany functionality.
- At least one ledger is mandatory in an implementation in which you record your accounting transactions.
- Business units are also mandatory because financial transactions are processed in business units.
- A shared service center is optional, but if used, must be a business unit.
- Divisions are optional and can be represented with a hierarchy of cost centers or by a second balancing segment value.
- Departments are mandatory because they track your employees.
- Optionally, add an item master organization and inventory organizations if you are tracking your inventory transactions in Oracle Fusion Applications.

---

**Note**

Some Oracle Fusion Human Capital Management and Customer Relationship Management implementations do not require recording of accounting transactions and therefore, do not require implementation of a ledger.

---

**Note**

The InFusion Corporation is a legal entity but is not discussed in this example.

---

## Define Common Financials Configuration for Rapid Implementation

### Define Financials and Fusion Accounting Hub Configuration for Rapid Implementation: Overview

Use the Define Financials and Fusion Accounting Hub Configuration for Rapid Implementation task list to streamline your setup configuration to focus only on the critical setup steps. The rapid implementation task lists minimize the time needed for you to complete your key setups and enable the day-to-day use of Oracle Fusion Financials and Oracle Fusion Accounting Hub applications.

To create an implementation project that includes the Define Financials and Fusion Accounting Hub Configuration for Rapid Implementation task list, use the Manage Implementation Projects page in the Setup and Maintenance work area. The application implementation manager can customize the task list and assign and track each task.

#### Task Lists

The Define Financials and Fusion Accounting Hub Configuration for Rapid Implementation task list contains the following task lists:

Task List	Description
Define Common Financials Configuration for Rapid Implementation	Define the common configuration for Oracle Fusion Financials rapid implementation, which includes enterprise structures and banks setup.
Define Transaction Taxes for Rapid Implementation	Configure tax setup according to local and international tax requirements.
Define Fusion Accounting Hub Configuration for Rapid Implementation	Define the configuration for Oracle Fusion Accounting Hub rapid implementation, which includes Oracle Fusion General Ledger, intercompany, and budgeting setup.
Define Invoicing and Payments Configuration for Rapid Implementation	Define invoicing and payments configuration for Oracle Fusion Payables rapid implementation.
Define Expenses Configuration for Rapid Implementation	Define corporate expense policies and rules, expense types, and expense report approval rules for Oracle Fusion Expenses rapid implementation.
Define Fixed Assets Configuration for Rapid Implementation	Define the configuration for Oracle Fusion Assets rapid implementation.
Define Receivables Configuration for Rapid Implementation	Define the configuration for Oracle Fusion Receivables rapid implementation.
Define Financials Security Configuration for Rapid Implementation	Define the configuration for user and data roles setup for Oracle Fusion Financials rapid implementation.

The rapid implementation task lists include tasks that are:

- Critical setup tasks for initial setup.
- Setup steps that are required by most users.

---

#### Note

You are not restricted to only the setup configuration in the rapid implementation task list. You can manually add the standard Financials or Fusion Accounting Hub offering task lists and tasks to your rapid implementation project to change and update your setup.

---

### Configuring Rapid Implementation Tasks Lists

Based on the applications you are implementing, you can streamline your task lists even more. For example, when you are only implementing:

- Oracle Fusion Accounting Hub: Delete the following task lists from your implementation project:
  - Define Transaction Taxes for Rapid Implementation
  - Define Invoicing and Payments Configuration for Rapid Implementation
  - Define Expenses Configuration for Rapid Implementation
  - Define Fixed Assets Configuration for Rapid Implementation
  - Define Receivables Configuration for Rapid Implementation

- Oracle Fusion Payables, Expenses, and Assets: Delete the Define Receivables Configuration for Rapid Implementation task lists from your implementation project:
- Oracle Fusion Receivables: Delete the following task lists from your implementation project:
  - Define Invoicing and Payments Configuration for Rapid Implementation
  - Define Expenses Configuration for Rapid Implementation
  - Define Fixed Assets Configuration for Rapid Implementation

## Creating an Oracle Fusion Financials Rapid Implementation Project: Worked Example

This example shows how to create an implementation project for the Oracle Fusion Financials and Oracle Fusion Accounting Hub rapid implementation task lists.

The following table summarizes key decisions for this scenario:

Decision to Consider	In This Example
What Oracle Fusion applications are included in this implementation?	The Oracle Fusion applications being implemented are: <ul style="list-style-type: none"> <li>• General Ledger</li> <li>• Subledger Accounting</li> <li>• Tax</li> <li>• Payables</li> <li>• Receivables</li> <li>• Expenses</li> <li>• Assets</li> </ul>
Are the setup requirements unique to this organization?	No
Can the rapid implementation task lists and tasks be used for this implementation?	Yes

### Creating the Implementation Project

1. On the Setup and Maintenance work area click the **Manage Implementation Projects** link.
2. Click **Create** in the Search Results table.
3. On the Create Implementation Project: Enter Basic Information page, enter **Implementation Project-FIN\_Rapid\_Implementation** in the **Name** field.
4. Click in the **Description** field to automatically update the name in the description.

5. Click the **Save and Open Project** button.
6. On the Implementation Project: Implementation Project-FIN\_Rapid\_Implementation page, click **Add** in the Task Lists and Tasks table.
7. On the **Select and Add: Task Lists and Task** dialog box, enter **%Rapid Implementation%** in the **Name** field. Use wild card characters if you do not know the exact name of the task list or task.
8. Click the **Search** button.
9. Select the Define Financials and Fusion Accounting Hub Configuration for Rapid Implementation row.
10. Click the **Done** button.
11. On the Implementation Project: Implementation Project-FIN\_Rapid\_Implementation page, expand the task list to see the task lists and tasks associated with your implementation project.

---

### **Important**

If you are not implementing all of the applications, you can delete the task lists associated with those applications. However, your rapid implementation project must always include the following task lists:

- Define Financials and Fusion Accounting Hub Configuration for Rapid Implementation
- Define Common Financials Configuration for Rapid Implementation
- Define Transaction Taxes for Rapid Implementation task list if you are not using Oracle Fusion Tax for your transaction tax processing.
- Define Fusion Accounting Hub Configuration for Rapid Implementation
- Define Financials Security Configuration for Rapid Implementation

The Open First Period task must also be included in all rapid implementation projects.

---

12. Click the **Done** button.

## **Enterprise Structures Rapid Implementation: Overview**

You can use Rapid Implementation spreadsheet template to rapidly implement Oracle Fusion enterprise structures and set up the following objects:

- Chart of accounts, including value sets, value set values, structures, and structure instances
- Account hierarchies including trees and tree versions
- Accounting calendars (monthly type only)
- Legal entities including legal addresses, jurisdictions, and identifiers

- Primary ledgers with legal entity assignments to primary balancing segment values
- Business units

Below are the Functional Setup Manager tasks which are required to be performed to rapidly implement the Oracle Fusion General Ledger:

- **Create Chart of Accounts, Ledger, Legal Entities, and Business Units in Spreadsheet:** Downloads the rapid implementation excel spreadsheet template.
- **Upload Chart of Accounts:** Invokes the **Create Enterprise Structure** process from the **Scheduled Process** page with the upload data file as a parameter. You need to upload your first file, ChartOfAccounts.xml, created using the template and monitor for the successful completion of the process.
- **Deploy Chart of Accounts:** Opens the **Manage Chart of Accounts Instance** page. Select the accounting flexfield and press the **Deploy** button to deploy the accounting flexfield.
- **Upload Ledger, Legal Entities, and Business Units:** Invokes the **Create Enterprise Structure** process from the **Scheduled Process** page with the upload data file as a parameter. You need to upload your second file, FinancialsCommonEntities.xml, created using the template and monitor for the successful completion of the process.

The following are best practices and recommendations:

- Design your chart of accounts carefully, including the number and sequence of segments.
- Identify the aspects of your business that you need to track and analyze, such as company, division, cost center, department, account, and line of business.
- Anticipate future growth and reporting requirements by defining one or more future segments. Assign those segments with a default value to streamline data entry.
- Create account hierarchies to reflect managerial, legal, or geographical relationships between your value set values. Your financial balances are preaggregated at each parent level in your hierarchy, thus allowing fast and robust account inquiry and drill down.

---

#### Note

Once you begin using your chart of accounts, making changes to the segments is neither recommended nor supported.

---

## Preparing and Loading Enterprise Structures Rapid Implementation Templates: Worked Example

Your company is implementing Oracle Fusion General Ledger using the rapid implementation methodology to create your enterprise structures.

Use the Microsoft Excel template to rapidly implement the following enterprise structures: chart of accounts, hierarchies, legal entities, accounting calendar, primary ledgers, and business units.

## Gathering the Data

Start your implementation carefully, by first gathering the necessary data:

1. Determine your chart of accounts segments.
2. Compile your segment values and hierarchies data.
3. Identify your legal entities and business units.
4. Determine your retained earnings account.

## Preparing the Template Data Sheets

Complete the data sheets for each enterprise structure.

1. Complete the **COA, Calendar, and Ledger** data sheet with your setup data for your chart of accounts, calendar, and ledger.

Entry Field	Comments
Ledger	Enter the name of primary ledger.
Ledger Currency	Select the primary ledger currency from the list of values.
Retained Earnings Account	Enter the retained earnings account. Ensure all the segment values are child values and postable. The value of natural account segment must be an account type of owner's equity.
Enable Average Balances	Select Yes to enable average balancing. Default value is No.
Fiscal Year Start Date	Enter the start date of the accounting calendar with the format mm/dd/yyyy.
Period Frequency	This value is monthly and read only.  <b>Note</b>  The rapid implementation spreadsheet only supports the creation of a monthly calendar. If you need a different type of calendar, use the standard Financials task list instead of the rapid implementation spreadsheet.
Adjusting Periods	Select the desired adjustments periods from the following list of values. <ul style="list-style-type: none"> <li>• Once at year end</li> <li>• Once at mid of year and once at year end</li> <li>• Quarterly</li> <li>• None</li> </ul>
Segment	Enter the segment name for each segment.

Segment Label	<p>Assign the following segment label from the list of values to your segments.</p> <ul style="list-style-type: none"> <li>• Primary, Second, and Third Balancing</li> <li>• Cost Center</li> <li>• Natural Account</li> <li>• Intercompany</li> <li>• Local Use</li> </ul> <hr/> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>• Primary Balancing and Natural Account are mandatory.</li> <li>• Assign segment labels once.</li> <li>• The Intercompany segment is optional. If specified, the segment reuses the value set created for the Primary Balancing segment.</li> <li>• The Second and Third Balancing are optional. Second balancing segment may be assigned with the Cost Center segment.</li> </ul>
Short Prompt	Enter the short prompt for each segment.
Display Width	Enter the display width you require for each segment.
Add Segment Sheets button	<p>After entering all your segments, press this button. A data sheet is created for each additional segment where you can enter the segment values and hierarchies.</p> <hr/> <p><b>Note</b></p> <p>The segment value and hierarchies data entry sheets for your Primary Balancing segment and Natural Account segment are provided.</p>

2. Complete the **Business Units** data entry sheet if needed.

Entry Field	Comments
Name	<p>Enter the name of your business units (BUs).</p> <hr/> <p><b>Note</b></p> <p>The process enables all the functions and defaults the following:</p> <ul style="list-style-type: none"> <li>• Reference Data Set is <b>Common</b>.</li> <li>• Primary Ledger created in this process is the default ledger for all BUs.</li> <li>• First entered legal entity on the <b>Companies and Legal Entities</b> data sheet is the default legal entity for all BUs.</li> </ul>

3. Complete the **Companies and Legal Entities** data entry sheet for your companies, hierarchies, and legal entity data.

Entry Field	Comments
Parent1 to Parent9	<p>To define hierarchies, enter the parent values in Parent1 to Parent9 column. First level parent is entered in the column Paent1. The parent of Parent1 is entered in the column Parent2 and so on.</p>
Child Value	<p>Enter all the child values or level0 values which are your postable account values in this column. No parent values are entered in this column.</p> <hr/> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>• Enter the child values of a parent first before you start entering the parent. Duplicate values are not allowed.</li> <li>• Enter a mandatory value and description on each row until the last row.</li> <li>• Plan the hierarchies before you start entering them in this template.</li> </ul>
Legal Entity Name	<p>This is a mandatory field.</p>

Legal Entity Identifier	This is a mandatory field.
Country	Select from the list of values provided for this mandatory field. The Identifying Jurisdiction is derived from the country you select.
Address Line	This is a mandatory field. Enter the address in comma separated format.
City, State, County, Province, and Postal Code	Address for the legal entity.
Legal Entity Registration Number	This is a mandatory field.
Legal Reporting Unit Registration Number	This is a mandatory field.  <b>Note</b>  Map the legal entity details to the primary balancing segment value. If the legal entity maps to multiple primary balancing segment values consecutively, only specify the legal entity for the first primary balancing segment value.

4. Complete the **Natural Accounts** data entry sheet if needed.

Entry Field	Comments
Parent1 to Parent9 and Child Value	Enter your parent values in these columns. First level parents are entered in the column Parent1 and so on. Enter the child values of a parent first before you start entering the parent. Duplicate values are not allowed.
Description	This is a mandatory field. Provide descriptions for each of the parent values as well as the child values.
Account Type	This is a mandatory field. Select from the following values: <ul style="list-style-type: none"> <li>• Asset</li> <li>• Expense</li> <li>• Liability</li> <li>• Owner's Equity</li> <li>• Revenue</li> </ul>

Financial Category	Not a mandatory field. The seeded values consist of the standard financial categories. For example: Accounts payable, Accounts receivable, Accrued liabilities, Accumulated depreciation, Cash, and Common stock.
--------------------	---

## Loading the Templates

1. After you finish preparing the data in the sheets, click the **Generate Chart of Accounts File** button on **COA, Calendar, and Ledger** sheet. The program generates an XML data file called **ChartOfAccounts.xml** for the entered chart of accounts and hierarchies setup data.
2. Save the file in a local directory.
3. Click the **Generate Ledger, LE, and BU File** button on **COA, Calendar, and Ledger** sheet. The program generates an XML data file called **FinancialsCommonEntities.xml** for the entered ledger, legal entities, and business units setup data.
4. Save the file in a local directory.
5. From the **Setup and Maintenance** page in the Functional Setup Manager, search for and select the **Upload Chart of Accounts** task. This task launches the **Upload Enterprise Structures** process.
6. Click on the **Upload File** button and select the first file you saved called **ChartOfAccounts.xml**.
7. Click the **Submit** button. The **Scheduled Processes** page opens.
8. Verify that the process completes without errors or warnings.
9. Press the **Done** button.
10. Select the **Deploy Chart of Accounts** task from the **Setup and Maintenance** page. Select the **Deploy** button for the Accounting Flexfield.
11. Refresh the page until the green check mark appears and verifies that the deployment was successful.
12. From **Setup and Maintenance** page select the task **Upload Ledger, Legal Entities and Business Units**. This launches the **Upload Enterprise Structures** process.
13. Click on the **Upload File** button and select the second xml file you saved called **FinancialsCommonEntities.xml**.
14. Click the **Submit** button. The **Scheduled Processes** page open
15. Verify that the process completed without errors or warnings.
16. Press **Done** button.

## Running Automatic Processes

1. Below are the processes that are run automatically during the rapid start flow to configure the ledger and associated setup and create the balances cubes.
  - Submit Account Configuration which includes general ledger balances cube creation

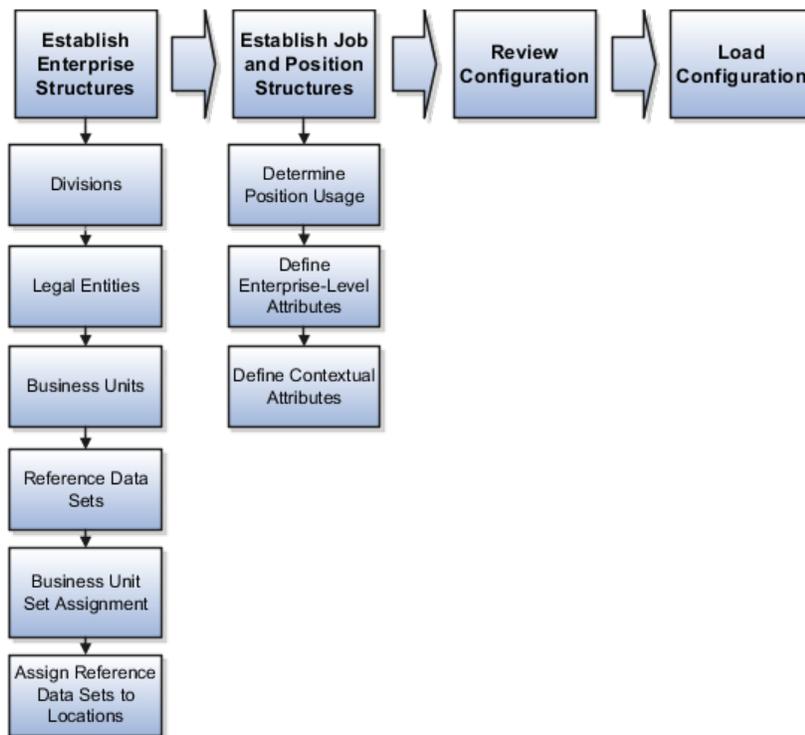
- Online Audit of Hierarchy Data of tree and tree versions
- Row and Column Flattening of hierarchy data
- Publish Hierarchies

## Define Initial Configuration with the Enterprise Structures Configurator

### Establishing Enterprise Structures Using the Enterprise Structures Configurator: Explained

The Enterprise Structures Configurator is an interview-based tool that guides you through the process of setting up a basic enterprise structure. By answering questions about your enterprise, the tool creates a structure of divisions, legal entities, business units, and reference data sets that reflects your enterprise structure. After you create your enterprise structure, you also follow a guided process to determine whether or not to use positions, and whether to set up additional attributes for jobs and positions. After you define your enterprise structure and your job and position structures, you can review them, make any necessary changes, and then load the final configuration.

This figure illustrates the process to configure your enterprise using the Enterprise Structures Configurator.



To be able to use the Enterprise Structures Configurator, you must select the Enterprise Structures Guided Flow feature for your offerings on the Configure

Offerings page in the Setup and Maintenance work area. If you do not select this feature, then you must set up your enterprise structure using individual tasks provided elsewhere in the offerings, and you cannot create multiple configurations to compare different scenarios.

## Establish Enterprise Structures

To define your enterprise structures, you use the guided flow within the Establish Enterprise Structures task to enter basic information about your enterprise, such as the primary industry and the location of your headquarters. You then create divisions, legal entities, business units, and reference data sets. The Establish Enterprise Structures task enables you to create multiple enterprise configurations so that you can compare different scenarios. Until you load a configuration, you can continue to create and edit multiple configurations until you arrive at one that best suits your enterprise.

## Establish Job and Position Structures

You also use a guided process to determine whether you want to use jobs only, or jobs and positions. The primary industry that you select in the Establish Enterprise Structures task provides the application with the information needed to make an initial recommendation. You can either accept the recommendation, or you can answer additional questions about how you manage people in your enterprise, and then make a selection. After you select whether to use jobs or positions, the guided process prompts you to set up a descriptive flexfield structure for jobs, and for positions if you have chosen to use them. Descriptive flexfields enable you to capture additional information when you create jobs and positions.

## Review Configuration

Finally, you can review a summary of the results of the two interview processes. For each configuration, the online summary lists the divisions, legal entities, business units, reference data sets, and job and position structures that the application will create when you load the configuration.

For a more detailed analysis of a configuration, you can access the Technical Summary Report. This report lists the same information as the online summary, but also lists the following information that will be created by the application when you load the configuration, based on your configuration:

- Legislative data groups (the application creates one legislative data group for each country that is identified in the configuration.)
- Name of the legislative data group that will be assigned to the payroll statutory unit that is generated for each legal entity.
- Organization hierarchy.

The Technical Summary report also lists the default settings that will be loaded for these fields, which you access from the Manage Enterprise HCM Information task: **Worker Number Generation**, **Employment Model** and **Allow Employment Terms Override**. You can print the Technical Summary Report for each of your configurations and compare each scenario.

---

### Note

If your PDF viewer preferences are set to open PDFs in a browser window, the Technical Summary report replaces the Oracle Fusion application. Use your browser's Back button to return to the application.

---

## **Load Configuration**

You can load only one configuration. When you load a configuration, the application creates the divisions, legal entities, business units, and so on. After you load the configuration, you then use individual tasks to edit, add, and delete enterprise structures.

## **Designing an Enterprise Configuration: Example**

This example illustrates how to set up an enterprise based on a global company operating mainly in the US and the UK with a single primary industry.

### **Scenario**

InFusion Corporation is a multinational enterprise in the high technology industry with product lines that include all the components that are required to build and maintain air quality monitoring (AQM) systems for homes and businesses. Its primary locations are in the US and the UK, but it has smaller outlets in France, Saudi Arabia, and the United Arab Emirates (UAE).

### **Enterprise Details**

In the US, InFusion employs 400 people and has a company revenue of \$120 million. Outside the US, InFusion employs 200 people and has revenue of \$60 million.

### **Analysis**

InFusion requires three divisions. The US division will cover the US locations. The Europe division will cover the UK and France. Saudi Arabia and the UAE will be covered by the Middle East division.

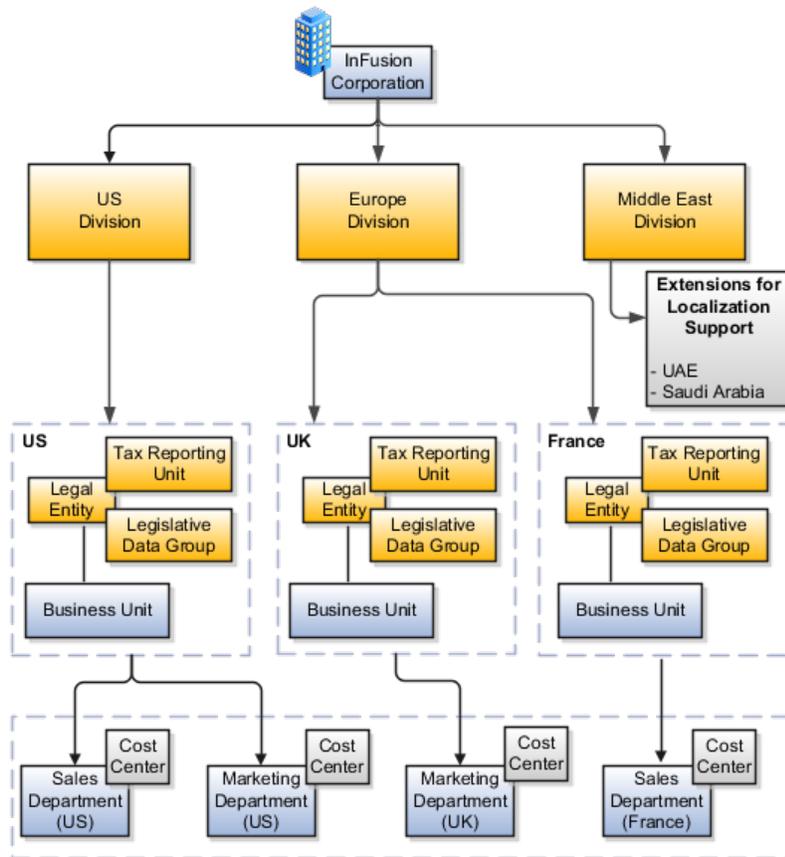
InFusion requires legal entities with legal employers, payroll statutory units, tax reporting units, and legislative data groups for the US, UK, France, Saudi Arabia, and UAE, in order to employ and pay its workers in those countries.

InFusion requires a number of departments across the enterprise for each area of business, such as sales and marketing, and a number of cost centers to track and report on the costs of those departments.

InFusion requires business units for human capital management (HCM) purposes. InFusion has general managers responsible for business units within each country. Those business units may share reference data. Some reference data can be defined within a reference data set that multiple business units may subscribe to. Business units are also required for financial purposes. Financial transactions are always processed within a business unit.

Based on this analysis, InFusion requires an enterprise with multiple divisions, ledgers, legal employers, payroll statutory units, tax reporting units, legislative data groups, departments, cost centers, and business units.

This figure illustrates the enterprise configuration that results from the analysis of InFusion Corporation.



## Division: Explained

Managing multiple businesses requires that you segregate them by their strategic objectives and measure their results. Responsibility to reach objectives can be delegated along the management structure. Although related to your legal structure, the business organizational hierarchies do not need to reflect directly the legal structure of the enterprise. The management entities and structure can include divisions and subdivisions, lines of business, and other strategic business units, and include their own revenue and cost centers. These organizations can be included in many alternative hierarchies and used for reporting, as long as they have representation in the chart of accounts.

### Divisions

A division refers to a business oriented subdivision within an enterprise, in which each division organizes itself differently to deliver products and services or address different markets. A division can operate in one or more countries, and can be comprised of many companies or parts of different companies that are represented by business units.

A division is a profit center or grouping of profit and cost centers, where the division manager is responsible for attaining business goals including profit goals. A division can be responsible for a share of the company's existing

product lines or for a separate business. Managers of divisions may also have return on investment goals requiring tracking of the assets and liabilities of the division. The division manager reports to a top corporate executive.

By definition a division can be represented in the chart of accounts. Companies may choose to represent product lines, brands, or geographies as their divisions: their choice represents the primary organizing principle of the enterprise. This may coincide with the management segment used in segment reporting.

Oracle Fusion Applications supports a qualified management segment and recommends that you use this segment to represent your hierarchy of business units and divisions. If managers of divisions have return on investment goals, make the management segment a balancing segment. Oracle Fusion applications allows up to three balancing segments. The values of the management segment can be comprised of business units that roll up in a hierarchy to report by division.

Historically, divisions were implemented as a node in a hierarchy of segment values. For example, Oracle E-Business Suite has only one balancing segment, and often the division and legal entity are combined into a single segment where each value stands for both division and legal entity.

### **Use of Divisions in Oracle Fusion Human Capital Management (HCM)**

Divisions are used in HCM to define the management organization hierarchy, using the generic organization hierarchy. This hierarchy can be used to create organization based security profiles.

## **Adding a New Division After Acquiring a Company: Example**

This example shows how to restructure your enterprise after acquiring a new division.

### **Scenario**

You are part of a senior management team at InFusion Corporation. InFusion is a global company with organizations in the United States (US), the United Kingdom (UK), France, China, Saudi Arabia, and the United Arab Emirates (UAE). Its main area of business is in the high tech industry, and it has just acquired a new company. You must analyze their current enterprise structure and determine what new organizations you need to create to accommodate the new company.

### **Details of the Acquired Company**

The acquired company is a financial services business based in Germany. Because the financial services business differs significantly from the high tech business, you want to keep the financial services company as a separate business with all the costs and reporting rolling up to the financial services division.

### **Analysis**

The following table summarizes the key decisions that you must consider when determining what new organizations to set up and how to structure the enterprise.

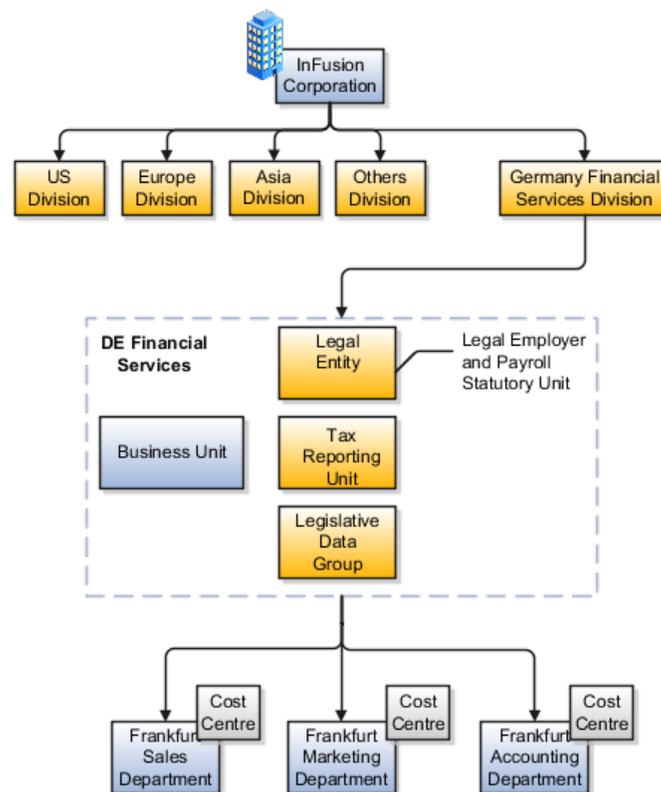
Decision to Consider	In This Example
Create location?	The financial services company is based in Frankfurt as are the departments, so you need to create only one location.
Create separate division?	Yes. Although the new division will exist within the current enterprise structure, you want to keep the financial services company as a separate line of business. Creating a separate division means you can manage the costs and reporting separately from the InFusion Corporation. It also means you do not have to modify any existing organizations in the enterprise setup.
Create business unit?	Yes. The financial services business requires you to create several jobs that do not exist in your high tech business. You can segregate the jobs that are specific to financial services in a new business unit.
How many departments?	The financial services company currently has three departments for sales, accounting, and marketing. As you have no plans to downsize or change the company, you can create three departments to reflect this structure.
How many cost centers?	Although you can have more than one cost center tracking the costs of a department, you decide to create one cost center for each department to track costs.
How many legal entities?	<p>Define a legal entity for each registered company or other entity recognized in law for which you want to record assets, liabilities, and income, pay transaction taxes, or perform intercompany trading. In this case, you need only one legal entity.</p> <p>You must define the legal entity as a legal employer and payroll statutory unit. As the new division operates in Germany only, you can configure the legal entity to suit Germany legal and statutory requirements.</p> <hr/> <p><b>Note</b></p> <p>When you identify the legal entity as a payroll statutory unit, the application transfers the legal reporting unit that is associated with that legal entity to Oracle Fusion HCM as a tax reporting unit.</p> <hr/>
Create legislative data group?	Yes. Because you currently do not employ or pay people in Germany, you must create one legislative data group to run payroll for the workers in Germany.

Based on the analysis, you must create the following:

- One new division
- One new location
- Three new departments
- Three new cost centers

- One new legal entity
- One new legislative data group

The following figure illustrates the structure of InFusion Corporation after adding the new division and the other organizations.



## Define Geographies

### Defining Address Cleansing: Explained

Address cleansing provides a way to validate, correct, and standardize addresses that are entered in a user interface. Geography validation only validates the geography attributes of an address, for example, State, City, and Postal codes; address cleansing validates both the geography attributes and the address line attributes.

Address cleansing can only be used through the Oracle Fusion Trading Community Data Quality product, because the feature is delivered using Data Quality integration. You need to ensure that you have a license for the countries that will use Trading Community Data Quality data cleansing.

You can specify the real time address cleansing level for each country by choosing either **None**, meaning that there is no real time address cleansing, or by choosing **Optional**, meaning that you will have the choice to cleanse addresses. Once you have enabled address cleansing for a country a **Verify Address** icon appears at address entry points in the application. You can then click the icon

to perform address cleansing and receive a corrected, standardized address. If Trading Community Data Quality does not find a matching address the application will alert you.

## Geography Structure, Hierarchy, and Validation: How They Fit Together

There are three components that are dependent on each other when defining a country: geography structure, geography hierarchy, and geography validation. Every country has to have the geography structure defined first before the hierarchy can be defined, and the geography hierarchy has to be defined before the validation can be defined.

### Geography Structure

Firstly, you need to create a geography structure for each country to define which geography types are part of the country structure, and how the geography types are hierarchically related within the country structure. For example, you can create geography types called State, City, and Postal Code. Then you can rank the State geography type as the highest level within the country, the City as the second level, and the Postal Code as the lowest level within the country structure. Geography structure can be defined using the **Manage Geographies** task, or can be imported using tasks in the **Define Geographies** activity.

### Geography Hierarchy

Once the geography structure is defined, the geographies for each geography type can be added to the hierarchy. For example, below the United States you can create a geography called California using a State geography type.

As part of managing the geography hierarchy you can view, create, edit, and delete the geographies for each geography type in the country structure. You can also add a primary and alternate name and code for each geography. A geography hierarchy can be created using the **Manage Geographies** task, or can be imported using tasks in the **Define Geographies** activity.

### Geography Validation

After defining the geography hierarchy, you need to specify the geography validations for the country. You can choose which address style formats you would like to use for the country, and for each selected address style format you can map geography types to address attributes. You can also select which geography types should be included in geography or tax validation, and which geography types will display in a list of values during address entry in other user interfaces. The geography validation level for the country, such as error or warning, can also be selected.

## Geography Structures: Explained

A geography structure is a hierarchical grouping of geography types for a country. For example, the geography structure for the United States is the geography type of State at the top, then followed by the County, then the City, and finally the Postal Code.

You can use the geography structure to establish:

- How geographies can be related
- The types of geographies you can define for the country

## How Geographies Can Be Related

You can determine how a country's geographies are hierarchically related by creating the hierarchy of the geography types in the geography structure. When you define a country's structure the country geography type is implicitly at the top of the geography structure, and the numbering of the subsequent levels start with 1 as the next geography level after country.

You must add a geography type as a level in the country structure before you can define a geography for that geography type in a country. For example, before defining the state of California, the State geography type must be added to the United States country structure. Only one geography type can be used for each level, you cannot define more than one geography type at the same level.

---

### Note

After you first define a country structure you can only add geography types below the current lowest level, and delete geography types without defined geographies.

---

To simplify the creation of a country structure you can copy a structure from another country, and then amend the geography type hierarchy for the country.

## The Types of Geographies You Can Define for the Country

The application provides you with a set of available master reference geography types. If required, you can create a geography type before adding it to the country structure. Each geography type is added below the current lowest level.

---

### Note

If you want to delete a geography type that is not at the lowest level in the country structure, then you have to delete the geography type level and all the levels below it.

---

A geography type that you create within the country structure can be used for other country structures as well.

## Geography Hierarchy: Explained

Geography hierarchy is a data model that lets you establish conceptual parent-child relationships between geographies. A geography, such as Tokyo or Peru, describes a boundary on the surface of the earth. The application can extrapolate information based on this network of hierarchical geographical relationships.

For example, in the geography hierarchy the state of California is defined as the parent of San Mateo county, which is the parent of Redwood City, which is

the parent of the postal code 94065. If you enter just 94065, the application can determine that the postal code is in California, or that the corresponding city is Redwood City.

The application leverages geography hierarchy information to facilitate business processes that rely on geography information, for example, tax calculation, order sourcing rules, sales territory definition. The geography hierarchy information is centrally located in the Trading Community Model and shared among other application offerings.

The top level of the geography hierarchy is Country, so the hierarchy essentially contains countries and their child geographies. Other aspects of the geography hierarchy include:

- Geography
- Geography type
- Geography usage
- Master reference geography hierarchy
- User defined zones

## **Geography**

A geography is a boundary such as a country, state, province or city. It is a physical space with boundaries that is a defined instance of a geography type. For example, San Jose is a geography of the City geography type.

## **Geography Type**

Geography types are a divisional grouping of geographies, which can be either geopolitical (for example, City, Province, and District) or user defined (for example, Continent, Country Regions, Tax Regions).

## **Geography Usage**

Geography usage indicates how a geography type or geography is used in the application. A master reference geography always has the usage of Master Reference. User defined zones can have the usages of Tax, Shipping, or Territory, based on what is relevant for their purpose.

## **Master Reference Geography Hierarchy**

The geography hierarchy data is considered to be the single source of truth for geographies. It is all the data, including geography types and geographies, that you define and maintain in the Trading Community Model tables.

The geography usage for the entire hierarchy is the master reference, and defined geography types and geographies are considered as master reference geography types and geographies. For example, Country is a universally recognized geography type, and United States is considered a master geography.

## User Defined Zones

User defined zones are a collection of geographical data, created from master reference data for a specific purpose. For example, territory zones are collections of master reference geographies ordered in a hierarchy. Tax and shipping zones are collections of master reference geographies without a hierarchical grouping.

## Geography Validation: Explained

Geography validation determines the geography mapping and validation for a country's address styles, as well as the overall geography validation control for a country.

The **No Styles Format** address style format is the default address style format for a country. By defining the mapping and validation for this format you will ensure that validations can be performed for any address in the country. After the **No Styles Format** is defined you can set up additional mapping for specific address styles.

For each address style format, you can define the following:

- Map to attribute
- Enable list of values
- Tax validation
- Geography validation
- Geography validation control

### Map to Attribute

For every address style format, you can map each geography type to an address attribute. For example, you can map the **State** geography type to the **State** address attribute for the United States, or map the **State** geography type to the **County** address attribute for the United Kingdom. The geography types that appear are based on how the country structure is defined. The list of address attributes that appear are based on address formats delivered with the application, or your customer defined address formats.

---

### Note

You only need to map geography types that you want to use for geography or tax validation purposes.

---

### Enable List of Values

Once a geography type is mapped to an attribute, then you can specify whether the geography type will appear in a list of values during address entry in user interfaces. It is very important to review carefully if you want to enable a list of values. You should only enable a list of values if you have sufficient geography data imported or created for that geography. Once you have enabled a list of

values for an address attribute, you can only select the geography data available for the geography type. This means that if a specific geography value is not available in the geography hierarchy, you cannot create an address with a different geography value.

### **Tax Validation**

You can also specify whether a geography type will be included in tax validation. For example, for the United States North America address style format you specify that County, State, and City are used for tax validation. This will mean that when a transaction involves an address with the North America address style, the address must have the correct county, state, and city combination based on the geography hierarchy data, to be considered valid for tax calculation.

### **Geography Validation**

You can specify whether a geography type will be included in geography validation. This will mean that, for example, when the user enters a United States address using the North America address style format, the address must have the correct country, state, and postal code combination based on geography hierarchy data to be considered geographically valid.

If an address element is mapped to a geography type, but not selected for geography validation usage, then during address entry suggested values will be provided for the address element, but the address element will not be validated.

---

#### **Note**

For either the tax or geography validation, do not skip more than one consecutive level unless you are certain that the selected geography types can uniquely identify geographies. For example, the United States country structure is: State, County, City, and Postal Code, and you want to select just State and Postal Code for geography or tax validation. However, for the combination of California and 94065, the city can be either Redwood Shores or Redwood City. In this case, you should also select at least the City geography type for geography or tax validation.

---

### **Geography Validation Control**

You can select the geography validation level for a country. Validation will check if the entered address maps to the geography hierarchy data available for the country, and the geography validation control determines whether you can save an address that did not pass validation during address entry. For example, if the validation level is **Error**, then an address cannot be saved if the values do not match the geography hierarchy data.

These are the geography validation levels you can choose:

- **Error** - only completely valid addresses can be saved, with all mandatory address elements entered.
- **No Validation** - all addresses can be saved including incomplete and invalid addresses.

Regardless of the result of validation, the validation process will try to map any address attribute to a geography of the country, and store any mapping it could establish based on the available data. This is called **Geography Name Referencing** and it is executed as part of validation. The result of this referencing is used in several business processes in the application to map an address to a specific geography or zone.

---

#### Note

The Geography Dimension value in territories is derived from sell-to addresses of sales accounts. To use geography dimensions in territories, ensure that the geography elements in addresses, such as state, city, and postal code, are validated. You can do so by enabling geography validation for each country using the Manage Geographies task. While doing so, ensure that at least one level in the geography hierarchy is enabled for geography validation. It is recommended that you enable geography validation for all geography levels that you intend to use for territory definition for each country. You can enable a list of values containing specific geography elements. This will help users search and select appropriate geography values during addresses entry and eliminate all possibilities of wrong address entry. You can also set geography validation control to Error in the Manage Geography Validation page. This ensures that users can only use valid geography elements in addresses. If you have already created addresses before setting up geography validation for a country, you must execute the Run Maintain Geography Name Referencing task for that country after enabling geography validation to ensure that all your geography elements are validated.

---

## Importing Geographies: Explained

A geography, such as Tokyo or Peru, describes a boundary on the surface of the earth. You can create new geographies by importing data through interface tables. There are two options for populating the interface tables: using the tool of your preference to load the data or using file-based data import. If you plan to provide the data details in a source file, use the file-based import feature. If you will populate the interface table directly, run the geography loader process to import the data. Having a good understanding of the import entity, interface table, and destination table will help you prepare your import data.

Consider the following when importing geographies:

- File-based import option
- Geography loader process option
- Import object entity, interface table, and destination tables

### File-Based Import Option

The file-based import process reads the data included in your XML or text file, populates the interface tables, and imports the data into the application destination tables. The **File-Based Data Import Setup and Maintenance** task list includes the tasks needed to configure the geography import object, create source file mappings, and schedule the import activities.

## Geography Loader Process Option

Populate the interface table with your import data, then navigate to the **Run Geography Loader Setup and Maintenance** task to schedule the import of data from the interface table to the destination table.

### Import Object Entity, Interface Table, and Destination Tables

The geography import object consists of one entity and interface table that forms the geography. If you are using file-based import, you can map your source file data to import entity attributes that correspond to the interface table columns. The import activity process populates the interface table based on the mapping and your source file. If using the geography loader scheduled process, populate the interface table directly using your preferred tool. If you need the unique IDs of existing application data for your import data, use the **Define Data Export Setup and Maintenance** task list to export the information.

---

#### Note

Spreadsheets containing detailed information about each interface table, including the import attributes, corresponding interface table columns, defaults, and validations, are available from the Oracle Enterprise Repository by searching on a specific interface table name or initiating a search using the **FusionApps: Interface Table** asset type.

---

The following lists the object entity, tables, and resulting application object:

File-Based Import Entities	Interface Tables	Destination Tables	Application Object
ImpGeography	HZ_IMP_GEOGRAPHIES	HZ_GEOGRAPHIES HZ_GEOGRAPHY_IDEN HZ_GEOGRAPHY_TYPE HZ_HIERARCHY_NODE	Geography

## Importing Geographies Using File-based Data Import: Worked Example

This example demonstrates how to import data using the File-Based Data Import tool. In this particular example you have a source file containing geography data that you want to import into the application, so that the geography data can be used for uses related to locations, such as real time address validation and tax purposes.

The following table summarizes the key decisions for this scenario:

Decisions to Consider	In This Example
What type of object are you importing?	Geography
What file type are you using for your source data?	Text file
Where are you uploading your source data file from?	Your desktop

What data type is your source data file?	Comma separated
Which fields are you importing into Oracle Fusion applications?	All, except for the RecordTypeCode field
When do you want to process the import?	Immediately

## Summary of the Tasks

These are the steps that are required to create an import activity and submit the import:

1. Determine what information is in the source file.
2. Create and schedule the import activity.
3. Monitor the import results.

## Prerequisites when importing additional geography data after your initial import

1. You need to ensure that the combination of Source ID and Parent Source ID values are unique for each row of data within a single import. However, your source data files do not need to have the same Source ID and Parent Source ID values as your previously imported geography data. If the geography structure levels and the parents for each geography value are the same, the changed IDs will not affect the import.
2. Ensure that all of the parents of a child geography are included in your data file so that the child geography can be added. For example, if you originally imported US, CA, and San Francisco, and now you want to import the city of San Jose in CA, then your data file needs to include US, CA, and San Jose.
3. Check that your source data file has the correct values for the geography data that you have already loaded. For example, if your initial import included the value US for country and CA as state, and in a subsequent import you have California as a state, your geography import will result in two state records (CA and California) in the application data, with the US as the country parent.

## Determine what information is in the source file

1. Your source geography data files should include a unique Source ID value for each row of data, and a Parent Source ID value which identifies the parent of that row of geography data. Source IDs, or Parent Source IDs, should not exceed 18 characters. An example of geography source data could be as follows:

Geography Level	Name	Source ID	Parent Source ID
1 (Country)	US	1	
2 (State)	CA	11	1
3 (County)	Alameda	111	11
4 (City)	Pleasanton	1111	111
4 (City)	Dublin	1112	111

## Create and schedule the import activity

You create an import activity, enter the import details, and schedule the import. An import activity definition provides the instructions for the import processing - this includes selecting the source file, or file location; mapping fields from the source file to the Oracle Fusion object and attribute; and scheduling the import.

1. Navigate to Setup and Maintenance and search for the Manage File Import Activities task. Click **Go to Task**.
2. In the Manage Import Activities page, click the **Create** icon.
3. In the Create Import Activity: Set Up page, create an import activity for the Geography object type by completing the fields, as shown in this table:

Field	Value
Name	Master Reference Geographies
Object	Geography
File Type	Text File
File Selection	Specific file
Upload From	Desktop
File Name	Choose relevant file from desktop
Data Type	Comma separated

---

### Note

Ensure that the file type that you select in the Create Import Activity: Set Up page matches the file type of the source data file.

---

4. Click **Next**.
5. On the Create Import Activity: Map Fields page, map each field from your source file to the Oracle Fusion object and attribute, as shown in this example:

Column Header	Example Value	Ignore	Object	Attribute
Primary Geography Name	Primary Geography Name	United States	Imp Geography	Primary Geography Name
Country Code	US	No	Imp Geography	Country Code
Record Type Code	0	Yes	Imp Geography	Record Type Code
Source ID	10265	No	Imp Geography	Source ID
Parent Source ID	1053	No	Imp Geography	Parent Source ID

If you do not want to import a column in the text file you can select **Ignore**.

---

**Note**

If you have any difficulties mapping the fields from your source file to the relevant Oracle Fusion applications object, you can use the import object spreadsheets for reference.

---

6. Click **Next**.
7. On the Create Import Activity: Create Schedule page, select Immediate in the Schedule field so that the import will start immediately.

Instead of immediately importing the data, you can choose a date and time to start the import. You can also specify if the import will be repeated, and the frequency of the repeated import.

8. Click **Next**.

## Monitor the import results

You monitor the progress of the Import Activity processing, and view completion reports for both successful records and errors.

1. On the Create Import Activity: Review and Activate page, you verify your import details in the Import Details, File Details, Import Options, and Schedule sections.
2. Your import details are correct so you click **Activate** to submit the import.

Once the import activity has completed, the Status field value will change to Completed.

## Importing and Exporting Territory Geography Zones: Explained

Territory geography zones are geographical boundaries that you can set up to replicate your organization's regions, such as a Pacific Northwest sales region. You can set up territory geography zones in one Oracle Fusion applications instance, and then after the territory geography zones are defined you can export the territory zones and import them into another Oracle Fusion applications instance.

To define your territory geography zones and then import your territory zones into another Oracle Fusion applications instance, you need to complete the following steps:

1. Import the master reference geography data into the Oracle Fusion application.
2. Define your territory geography zones using the Manage Territory Geographies task.
3. Export the territory geography zones.
4. Import the territory geography zones into another Oracle Fusion applications instance.

## Import the master reference geography data

Firstly, you need to import the master reference geography data. Master reference geography data consists of geography elements such as country, state, and city, and is required for any geographical information you store in the application, such as address information used in customer and sales records. For more information, refer to the Geography Hierarchy: Explained topic listed in the related topics section. Master reference geography data can be imported into the application using the Manage File Import Activities task in Setup and Maintenance - refer to the Importing Master Reference Geography Data: Worked Example topic listed in the related topics section for more information.

## Define your territory geography zones

Once the master reference geography data has been imported, you can then create your territory geography zones in the application using the Manage Territory Geographies task in Setup and Maintenance. For more information, refer to the Managing Territory Geographies: Worked Example topic listed in the related topics section.

## Export the territory geography zones

Once you have completed importing the master reference geography data and defining your territory geography zone tasks, you can create a configuration package to export the territory zone data. For more information, refer to the Exporting Setup Data demo listed in the related topics section.

## Import the territory geography zones

Once you have downloaded your configuration package for your territory geography zone setup, you can import the territory zones into another Oracle Fusion application instance. For more information, refer to the Importing Setup Data listed in the related topics section.

---

### Note

Ensure that you import your master reference geography data into the new Oracle Fusion instance before you import the configuration package.

---

## Managing Geography Structures, Hierarchies, and Validation: Worked Example

This example shows how to start the configuration of the geography structure, hierarchy, and validation for the country geography of the United Kingdom.

The following table summarizes the key decisions for this scenario.

Decisions to Consider	In This Example
Copy an existing country structure?	No, create a new country structure.

What is the structure of the geography types?	Create geography types with the following ranking structure:  1. County 2. Post Town
What is the geography hierarchy?	Create the following hierarchy:  1. Country of United Kingdom 2. County of Berkshire 3. Post Town of Reading
Which address style format will you use when mapping geography validations?	The default address style format, called the No Styles Format.
Are you using Oracle Fusion Tax for tax purposes?	No, do not select <b>Tax Validation</b> for the geography types.

Add the County and Post Town geography types to the geography structure. Then add the geographies for the County and Post Town geography types to define the geography hierarchy. Finally, specify the geography validations for the geography types you have added to the geography structure.

### Defining the geography structure

You add the County and Post Town geography types to the United Kingdom geography structure.

1. On the Manage Geographies page, enter GB in the **Code** field. Click **Search**.
2. On the Manage Geographies page, click **Structure Defined**.
3. On the Manage Geography Structure page, click the **Create** button next to the **Copy Country Structure From** field.
4. In the Geography Structure section, select the County list item in the **Add Geography Type** field.
5. Click **Add**.
6. Select the Post Town list item in the **Add Geography Type** field.
7. Click **Add**.

### Defining the geography hierarchy

You want to begin to create the geography hierarchy for the United Kingdom, so you add the geographies for the County and Post Town geography types using the geography hierarchy User Interfaces. You can also use the Manage File Import Activities task to import geography hierarchies using a csv or xml file.

1. On the Manage Geographies page, enter GB in the **Code** field. Click **Search**.
2. On the Manage Geographies page, click **Hierarchy Defined**.
3. On the Manage Geography Hierarchy page, Geography Hierarchy section, click the United Kingdom to highlight the table row.
4. Click the **Create** button.
5. In the **Create County** page, Primary and Alternate Names section, enter Berkshire in the **Name** field.

6. Click **Save and Close**.
7. On the Manage Geography Hierarchy page, Geography Hierarchy section, click Berkshire to highlight the table row.
8. Click the **Create** button.
9. In the **Create Post Town** page, Primary and Alternate Names section, enter Reading in the **Name** field.
10. Click Save and Close.

## Defining the geography validations

Now you want to specify the geography validations for the geography types you have added to the United Kingdom. You define the geography mapping and validation for the United Kingdom default address style format. You map the geography types to attributes, enable the geography types for Lists of Values and Geography validation, and set the geography validation level.

1. On the Manage Geographies page, click **Validation Defined**.
2. On the Manage Geography Validation page, Address Style section, click **No Styles Format** to highlight the table row.
3. For the County geography type, click the **County** list item in the **Map to Attribute** field.
4. Click the **Enable List of Values** option for the County geography type.
5. Click the **Geography Validation** option for the County geography type.
6. For the Post Town geography type, click the **City** list item in the **Map to Attribute** field.
7. Click the **Geography Validation** option for the Post Town geography type.
8. In the Geography Validation Control section, click the **Error** list item in the **Geography Validation Level for Country** field.
9. Click **Save and Close**.

## FAQs for Define Geographies

### When do I define address cleansing?

When address data entered into the application needs to conform to a particular format, in order to achieve consistency in the representation of addresses. For example, making sure that the incoming data is stored following the correct postal address format.

### Why can't I update a geography structure by copying an existing country structure?

You can only update a geography structure by adding existing geography types, or by creating new geography types and then adding them to the geography structure. You can only copy an existing country structure when you are defining a new country structure.

### **Why can't I delete a level of the country geography structure?**

If a geography exists for a country geography structure level then you cannot delete the level. For example, if a state geography has been created for the United States country geography structure, then the State level cannot be deleted in the country geography structure.

### **Can I add any geography to the geography hierarchy?**

Yes. However, the geography type for the geography that you want to add must be already added to the country geography structure.

### **Can I edit a specific geography in the geography hierarchy?**

Yes. In the Manage Geography Hierarchy page you can edit details such as the geography's date range, primary and alternate names and codes, and parent geographies.

### **How can I add a geography that is the level below another geography in a geography hierarchy?**

Select the geography that you want your geography to be created below, and then click the **Create** icon. This will allow you to create a geography for a geography type that is the level below the geography type you selected. The structure of the country's geography types are defined in the Manage Geography Structure page.

## **Define Legal Jurisdictions and Authorities**

### **Jurisdictions and Legal Authorities: Explained**

You are required to register your legal entities with legal authorities in the jurisdictions where you conduct business. Register your legal entities as required by local business requirements or other relevant laws. For example, register your legal entities for tax reporting to report sales taxes or value added taxes.

#### **Legal Jurisdictions and Authorities**

Define jurisdictions and related legal authorities to support multiple legal entity registrations, which are used by Oracle Fusion Tax and Oracle Fusion Payroll. When you first create a legal entity, the Oracle Fusion Legal Entity Configurator automatically creates one legal reporting unit for that legal entity with a registration.

### **Jurisdictions: Explained**

Jurisdiction is a physical territory such as a group of countries, country, state, county, or parish where a particular piece of legislation applies. French Labor Law, Singapore Transactions Tax Law, and US Income Tax Laws are examples of

particular legislation that apply to legal entities operating in different countries' jurisdictions. Judicial authority may be exercised within a jurisdiction.

Types of jurisdictions are:

- Identifying Jurisdiction
- Income Tax Jurisdiction
- Transaction Tax Jurisdiction

### **Identifying Jurisdiction**

For each legal entity, select an identifying jurisdiction. An identifying jurisdiction is your first jurisdiction you must register with to be allowed to do business in a country. If there is more than one jurisdiction that a legal entity needs to register with to commence business, select one as the identifying jurisdiction. Typically the identifying jurisdiction is the one you use to uniquely identify your legal entity.

Income tax jurisdictions and transaction tax jurisdictions do not represent the same jurisdiction. Although in some countries, the two jurisdictions are defined at the same geopolitical level, such as a country, and share the same legal authority, they are two distinct jurisdictions.

### **Income Tax Jurisdiction**

Create income tax jurisdictions to properly report and remit income taxes to the legal authority. Income tax jurisdictions by law impose taxes on your financial income generated by all your entities within their jurisdiction. Income tax is a key source of funding that the government uses to fund its activities and serve the public.

### **Transaction Tax Jurisdiction**

Create transaction tax jurisdictions through Oracle Fusion Tax in a separate business flow, because of the specific needs and complexities of various taxes. Tax jurisdictions and their respective rates are provided by suppliers and require periodic maintenance. Use transaction tax jurisdiction for legal reporting of sales and value added taxes.

## **Legal Authorities: Explained**

A legal authority is a government or legal body that is charged with powers to make laws, levy and collect fees and taxes, and remit financial appropriations for a given jurisdiction.

### **Legal Authorities**

For example, the Internal Revenue Service is the authority for enforcing income tax laws in United States. In some countries, such as India and Brazil, you are required to print legal authority information on your tax reports. Legal authorities are defined in the Oracle Fusion Legal Entity Configurator. Tax authorities are a subset of legal authorities and are defined using the same setup flow.

Legal authorities are not mandatory in Oracle Fusion Human Capital Management (HCM), but are recommended and are generally referenced on statutory reports.

## Creating Legal Jurisdictions, Addresses and Authorities: Examples

Define legal jurisdictions and related legal authorities to support multiple legal entity registrations, which are used by Oracle Fusion Tax and Oracle Fusion Payroll.

### Legal Jurisdictions

Create a legal jurisdiction by following these steps:

1. Navigate to the **Manage Legal Jurisdictions** page from the **Setup and Maintenance** work area by querying on the **Manage Legal Jurisdictions** task and selecting **Go to Task**.
2. Select **Create**.
3. Enter a unique **Name**, United States Income Tax.
4. Select a **Territory**, United States.
5. Select a **Legislative Category**, Income tax.
6. Select **Identifying**, Yes. Identifying indicates the first jurisdiction a legal entity must register with to do business in a country.
7. Enter a **Start Date** if desired. You can also add an **End Date** to indicate a date that the jurisdiction may no longer be used.
8. Select a **Legal Entity Registration Code**, EIN or TIN.
9. Select a **Legal Reporting Unit Registration Code**, Legal Reporting Unit Registration Number.
10. Optionally enter one or more **Legal Functions**.
11. Select **Save and Close**.

### Legal Addresses for Legal Entities and Reporting Units

Create a legal address for legal entities and reporting units by following these steps:

1. Navigate to the **Manage Legal Address** page from the **Setup and Maintenance** work area by querying on the **Manage Legal Address** task and selecting **Go to Task**.
2. Select **Create**.
3. Select **Country**.
4. Enter **Address Line 1**, Oracle Parkway.
5. Optionally enter **Address Line 2**, and **Address Line 3**.
6. Enter or Select **Zip Code**, 94065.
7. Select **Geography** 94065 and **Parent Geography** Redwood Shores, San Mateo, CA.

8. Optionally enter a **Time Zone**, US Pacific Time.
9. Select **OK**.
10. Select **Save and Close**.

## Legal Authorities

Create a legal authority by following these steps:

1. Navigate to the **Manage Legal Authorities** page from the **Setup and Maintenance** work area by querying on the **Manage Legal Authorities** task and selecting **Go to Task**.
2. Enter the **Name**, California Franchise Tax Board.
3. Enter the **Tax Authority Type**, Reporting.

---

### Note

Create an address for the legal authority.

- 
4. Select **Create**.
  5. The **Site Number** is automatically assigned.
  6. Optionally enter a **Mail Stop**.
  7. Select **Country**, United States
  8. Enter **Address Line 1**, 121 Spear Street, Suite 400.
  9. Optionally enter **Address Line 2**, and **Address Line 3**.
  10. Enter or Select **Zip Code**, 94105.
  11. Select **Geography** 94105 and **Parent Geography** San Francisco, San Francisco, CA.
  12. Select **OK**.
  13. Optionally enter a **Time Zone**, US Pacific Time.
  14. Optionally click the **One-Time Address** check box.
  15. The **From Date** defaults to today's date. Update if necessary.
  16. Optionally enter a **To Date** to indicate the last day the address can be used.

---

### Note

You can optionally enter **Address Purpose** details.

- 
17. Select **Add Row**.
  18. Select **Purpose**.
  19. The **Purpose from Date** will default to today's date.
  20. Optionally enter a **Purpose to Date**.
  21. Select **OK**.
  22. Select **Save and Close**.

## Creating Legal Entities, Registrations, and Reporting Units: Examples

Define a legal entity for each registered company or other entity recognized in law for which you want to record assets, liabilities, and income, pay transaction taxes, or perform intercompany trading.

### Legal Entity

From within an implementation project, create a legal entity by following these steps:

---

#### Note

Working within an implementation project is required because you select a scope value within an implementation project. The scope value is the legal entity that you will create or select to work within for your implementation project.

- 
1. Navigate to an implementation project that contains the **Define Legal Entities** task list from the **Setup and Maintenance** work area.
  2. Select **Go to Task** for the **Define Legal Entities** task list within the implementation project.

---

#### Note

The following message appears:

You must first select a scope value to perform the task.

- Select and add an existing scope value to the implementation project.
- Create a new scope value and then add it to the implementation project.

- 
3. Select **Create New**.
  4. From the **Manage Legal Entities** page select **Create**.
  5. Accept the default **Country**, United States.
  6. Enter **Name**, InFusion USA West.
  7. Enter **Legal Entity Identifier**, US0033.
  8. Optionally enter **Start Date**. When the start date is blank the legal entity is effective from the creation date.
  9. Optionally enter an **End Date**.
  10. Optionally, if your legal entity should be registered to report payroll tax and social insurance, select the **Payroll statutory unit** check box.
  11. Optionally, if your legal entity has employees, select the **Legal employer** check box.
  12. Optionally, if this legal entity is not a payroll statutory unit, select an existing payroll statutory unit to report payroll tax and social instance on behalf of this legal entity.

---

**Note**

Enter the **Registration Information**.

---

13. Accept the default **Identifying Jurisdiction**, United States Income Tax.
  14. Search for and select a **Legal Address**, 500 Oracle Parkway, Redwood Shores, CA 94065.
- 

**Note**

The legal address must have been entered previously using the **Manage Legal Address** task.

---

15. Select **OK**.
16. Optionally enter a **Place of Registration**.
17. Enter the **EIN or TIN**.
18. Enter the **Legal Reporting Unit Registration Number**.
19. Select **Save and Close** to navigate back to the Manage Legal Entities page.
20. Select **Done** to return to your implementation project. An issue with the done button has been fixed in 11g Release 1 (11.1.4).
21. In the **Legal Entity** choice list in the implementation project (just below the implementation project name and code), click **Select and Add Legal Entity** to choose the legal entity that you just created, and set the scope for the remainder of your setup.
22. Search for and select your legal entity from the **Manage Legal Entities** page.
23. Select **Save and Close**.

This sets the scope for your task list to the selected legal entity, as indicated in the **Legal Entity** choice list above the **Tasks and Task Lists** table.

## Legal Entity Registrations

A legal entity registration with the same name as that of the legal entity will be created by default. To verify this, locate the **Manage Legal Entity Registrations** task and then select **Go to Task**. To create another registration for the legal entity follow these steps:

1. Navigate to your implementation project from the **Setup and Maintenance** work area. Verify that the parent **Legal Entity** scope value is set correctly.
2. Expand the **Define Legal Entities** task list within the implementation project.
3. Select **Manage Legal Entity Registrations Go to Task**.
4. Select **Create**.
5. Enter **Jurisdiction**.

6. Enter **Registered Address**.
7. Enter **Registered Name**.
8. Optionally enter **Alternate Name, Registration Number, Place of Registration, Issuing Legal Authority, and Issuing Legal Authority Address, Start Date, and End Date**.
9. **Save and Close**.

### Legal Reporting Unit

When a legal entity is created, a legal reporting unit with the same name as that of the entity is also automatically created. To create more legal reporting units or modify the settings follow these steps:

1. Navigate to your implementation project from the **Setup and Maintenance** work area. Verify that the parent **Legal Entity** scope value is set correctly.
2. Select **Go to Task** for the **Define Legal Entities** task list within the implementation project.
3. Select **Create**.
4. Enter **Territory**, United States.
5. Enter **Name**.
6. Optionally enter a **Start Date**.

---

#### Note

Enter **Registration Information**.

- 
7. Search for and select **Jurisdiction**.

---

#### Note

Enter **Main Legal Reporting Unit** information.

- 
8. Select the value Yes or No for the **Main Legal Reporting Unit**. Set value to yes only if you are creating a new main (primary) legal reporting unit.
  9. Enter the **Main Effective Start Date**, 1/1/11.
  10. **Save and Close**.

## Legal Entities: Explained

A legal entity is a recognized party with rights and responsibilities given by legislation.

Legal entities have the right to own property, the right to trade, the responsibility to repay debt, and the responsibility to account for themselves to regulators, taxation authorities, and owners according to rules specified in the relevant legislation. Their rights and responsibilities may be enforced through the judicial system. Define a legal entity for each registered company or other entity

recognized in law for which you want to record assets, liabilities, expenses and income, pay transaction taxes, or perform intercompany trading.

A legal entity has responsibility for elements of your enterprise for the following reasons:

- Facilitating local compliance
- Taking advantage of lower corporation taxation in some jurisdictions
- Preparing for acquisitions or disposals of parts of the enterprise
- Isolating one area of the business from risks in another area. For example, your enterprise develops property and also leases properties. You could operate the property development business as a separate legal entity to limit risk to your leasing business.

### **The Role of Your Legal Entities**

In configuring your enterprise structure in Oracle Fusion Applications, you need to understand that the contracting party on any transaction is always the legal entity. Individual legal entities own the assets of the enterprise, record sales and pay taxes on those sales, make purchases and incur expenses, and perform other transactions.

Legal entities must comply with the regulations of jurisdictions, in which they register. Europe now allows for companies to register in one member country and do business in all member countries, and the US allows for companies to register in one state and do business in all states. To support local reporting requirements, legal reporting units are created and registered.

You are required to publish specific and periodic disclosures of your legal entities' operations based on different jurisdictions' requirements. Certain annual or more frequent accounting reports are referred to as statutory or external reporting. These reports must be filed with specified national and regulatory authorities. For example, in the United States (US), your publicly owned entities (corporations) are required to file quarterly and annual reports, as well as other periodic reports, with the Securities and Exchange Commission (SEC), who enforces statutory reporting requirements for public corporations.

Individual entities privately held or held by public companies do not have to file separately. In other countries, your individual entities do have to file in their own name, as well as at the public group level. Disclosure requirements are diverse. For example, your local entities may have to file locally to comply with local regulations in a local currency, as well as being included in your enterprise's reporting requirements in different currency.

A legal entity can represent all or part of your enterprise's management framework. For example, if you operate in a large country such as the United Kingdom or Germany, you might incorporate each division in the country as a separate legal entity. In a smaller country, for example Austria, you might use a single legal entity to host all of your business operations across divisions.

## **Creating Legal Entities in the Enterprise Structures Configurator: Points to Consider**

Using the Enterprise Structures Configurator (ESC), you can create legal entities for your enterprise automatically, based on the countries in which divisions

of your business operate, or you can upload a list of legal entities from a spreadsheet.

### Automatically Creating Legal Entities

If you are not certain of the number of legal entities that you need, you can create them automatically. To use this option, you first identify all of the countries in which your enterprise operates. The application opens the Map Divisions by Country page, which contains a matrix of the countries that you identified, your enterprise, and the divisions that you created. You select the check boxes where your enterprise and divisions intersect with the countries to identify the legal entities that you want the application to create. The enterprise is included for situations where your enterprise operates in a country and acts on behalf of several divisions within the enterprise and is a legal employer in a country. If you select the enterprise for a country, the application creates a country holding company.

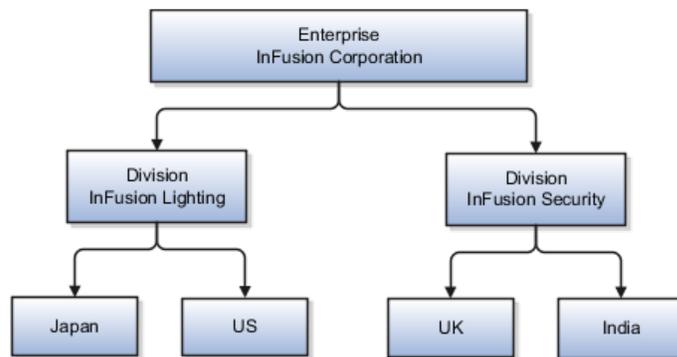
The application automatically creates the legal entities that you select, and identifies them as payroll statutory units and legal employers. For each country that you indicated that your enterprise operates in, and for each country that you created a location for, the application also automatically creates a legislative data group.

Any legal entities that you create automatically cannot be deleted from the Create Legal Entities page within the Enterprise Structures Configurator. You must return to the Map Divisions by Country page and deselect the legal entities that you no longer want.

### Example: Creating Legal Entities Automatically

InFusion Corporation is using the ESC to set up their enterprise structure. They have identified two divisions, one for Lighting, and one for Security. The Lighting division operates in Japan and the US, and the Security division operates in the UK and India.

This figure illustrates InFusion Corporation's enterprise structure.



This table represents the selections that InFusion Corporation makes when specifying which legal entities to create on the Map Divisions by Country page.

Country	Enterprise	InFusion Lighting	InFusion Security
Japan	No	Yes	No

US	No	Yes	No
UK	No	No	Yes
India	No	No	Yes

Based on the selections made in the preceding table, the ESC creates the following four legal entities:

- InFusion Lighting Japan LE
- InFusion Lighting US LE
- InFusion Security UK LE
- InFusion Security India LE

### **Creating Legal Entities Using a Spreadsheet**

If you have a list of legal entities already defined for your enterprise, you can upload them from a spreadsheet. To use this option, you first download a spreadsheet template, then add your legal entity information to the spreadsheet, and then upload directly to your enterprise configuration. You can export and import the spreadsheet multiple times to accommodate revisions.

## **Legal Entity in Oracle Fusion: Points to Consider**

Oracle Fusion Applications support the modeling of your legal entities. If you make purchases from or sell to other legal entities, define these other legal entities in your customer and supplier registers, which are part of the Oracle Fusion Trading Community Architecture. When your legal entities are trading with each other, you represent both of them as legal entities and also as customers and suppliers in your customer and supplier registers. Use legal entity relationships to determine which transactions are intercompany and require intercompany accounting. Your legal entities can be identified as legal employers and therefore, are available for use in Human Capital Management (HCM) applications.

There are several decisions that need to be considered in creating your legal entities.

- The importance of legal entity in transactions
- Legal entity and its relationship to business units
- Legal entity and its relationship to divisions
- Legal entity and its relationship to ledgers
- Legal entity and its relationship to balancing segments
- Legal entity and its relationship to consolidation rules
- Legal entity and its relationship to intercompany transactions
- Legal entity and its relationship to worker assignments and legal employer

- Legal entity and payroll reporting
- Legal reporting units

## **The Importance of Legal Entity in Transactions**

All of the assets of the enterprise are owned by individual legal entities. Oracle Fusion Financials allow your users to enter legal entities on transactions that represent a movement in value or obligation.

For example, the creation of a sales order creates an obligation for the legal entity that books the order to deliver the goods on the acknowledged date, and an obligation of the purchaser to receive and pay for those goods. Under contract law in most countries, damages can be sought for both actual losses, putting the injured party in the same state as if they had not entered into the contract, and what is called loss of bargain, or the profit that would have made on a transaction.

In another example, if you revalued your inventory in a warehouse to account for raw material price increases, the revaluation and revaluation reserves must be reflected in your legal entity's accounts. In Oracle Fusion Applications, your inventory within an inventory organization is managed by a single business unit and belongs to one legal entity.

## **Legal Entity and Its Relationship to Business Units**

A business unit can process transactions on behalf of many legal entities. Frequently, a business unit is part of a single legal entity. In most cases the legal entity is explicit on your transactions. For example, a payables invoice has an explicit legal entity field. Your accounts payables department can process supplier invoices on behalf of one or many business units.

In some cases, your legal entity is inferred from your business unit that is processing the transaction. For example, your business unit A agrees on terms for the transfer of inventory to your business unit B. This transaction is binding on your default legal entities assigned to each business unit. Oracle Fusion Procurement, Oracle Fusion Projects, and Oracle Fusion Supply Chain applications rely on deriving the legal entity information from the business unit.

## **Legal Entity and Its Relationship to Divisions**

The division is an area of management responsibility that can correspond to a collection of legal entities. If desired, you can aggregate the results for your divisions by legal entity or by combining parts of other legal entities. Define date-effective hierarchies for your cost center or legal entity segment in your chart of accounts to facilitate the aggregation and reporting by division. Divisions and legal entities are independent concepts.

## **Legal Entity and Its Relationship to Ledgers**

One of your major responsibilities is to file financial statements for your legal entities. Map legal entities to specific ledgers using the Oracle Fusion General

Ledger Accounting Configuration Manager. Within a ledger, you can optionally map a legal entity to one or more balancing segment values.

## **Legal Entity and Its Relationship to Balancing Segments**

Oracle Fusion General Ledger supports up to three balancing segments. Best practices recommend that one of these segments represents your legal entity to ease your requirement to account for your operations to regulatory agencies, tax authorities, and investors. Accounting for your operations means you must produce a balanced trial balance sheet by legal entity. If you account for many legal entities in a single ledger, you must:

1. Identify the legal entities within the ledger.
2. Balance transactions that cross legal entity boundaries through intercompany transactions.
3. Decide which balancing segments correspond to each legal entity and assign them in Oracle Fusion General Ledger Accounting Configuration Manager. Once you assign one balancing segment value in a ledger, then all your balancing segment values must be assigned. This recommended best practice facilitates reporting on assets, liabilities, and income by legal entity.

Represent your legal entities by at least one balancing segment value. You may represent it by two or three balancing segment values if more granular reporting is required. For example, if your legal entity operates in multiple jurisdictions in Europe, you might define balancing segment values and map them to legal reporting units. You can represent a legal entity by more than one balancing segment value, do not use a single balancing segment value to represent more than one legal entity.

In Oracle Fusion General Ledger, there are three balancing segments. You can use separate balancing segments to represent your divisions or strategic business units to enable management reporting at the balance sheet level for each division or business unit. For example, use this solution to empower your business unit and divisional managers to track and assume responsibility for their asset utilization or return on investment. Using multiple balancing segments is also useful when you know at the time of implementation that you are disposing of a part of a legal entity and need to isolate the assets and liabilities for that entity.

---

### **Note**

Implementing multiple balancing segments requires every journal entry that is not balanced by division or business unit, to generate balancing lines. Also, you cannot change to multiple balancing segments easily after you have begun to use the ledger because your historical data is not balanced by the new multiple balancing segments. Restating historical data must be done at that point.

To use this feature for disposal of a part of a legal entity, implement multiple balancing segments at the beginning of the legal entity's corporate life or on conversion to Oracle Fusion.

---

If you decided to account for each legal entity in a separate ledger, there is no requirement to identify the legal entity with a balancing segment value within the ledger.

---

**Note**

While transactions that cross balancing segments don't necessarily cross legal entity boundaries, all transactions that cross legal entity boundaries must cross balancing segments. If you make an acquisition or are preparing to dispose of a portion of your enterprise, you may want to account for that part of the enterprise in its own balancing segment even if it is not a separate legal entity. If you do not map legal entities sharing the same ledger to balancing segments, you will not be able to distinguish them using the intercompany functionality or track their individual equity.

---

**Legal Entity and Its Relationship to Consolidation Rules**

In Oracle Fusion Applications you can map legal entities to balancing segments and then define consolidation rules using your balancing segments. You are creating a relationship between the definition of your legal entities and their role in your consolidation.

**Legal Entity and its Relationship to Intercompany Transactions**

Use Oracle Fusion Intercompany functionality for automatic creation of intercompany entries across your balancing segments. Intercompany processing updates legal ownership within the enterprise's groups of legal entities. Invoices or journals are created as needed. To limit the number of trading pairs for your enterprise, set up intercompany organizations and assign them to your authorized legal entities. Define processing options and intercompany accounts to use when creating intercompany transactions and to assist in consolidation elimination entries. These accounts are derived and automatically entered on your intercompany transactions based on legal entities assigned to your intercompany organizations.

Intracompany trading, in which legal ownership isn't changed but other organizational responsibilities are, is also supported. For example, you can track assets and liabilities that move between your departments within your legal entities by creating departmental level intercompany organizations.

---

**Note**

In the Oracle Fusion Supply Chain applications, model intercompany relationships using business units, from which legal entities are inferred.

---

**Legal Entity and Its Relationship to Worker Assignments and Legal Employer**

Legal entities that employ people are called legal employers in the Oracle Fusion Legal Entity Configurator. You must enter legal employers on worker assignments in Oracle Fusion HCM.

**Legal Entity and Payroll Reporting**

Your legal entities are required to pay payroll tax and social insurance such as social security on your payroll. In Oracle Fusion Applications, you can register

payroll statutory units to pay and report on payroll tax and social insurance on behalf of many of your legal entities. As the legal employer, you might be required to pay payroll tax, not only at the national level, but also at the local level. You meet this obligation by establishing your legal entity as a place of work within the jurisdiction of a local authority. Set up legal reporting units to represent the part of your enterprise with a specific legal reporting obligation. You can also mark these legal reporting units as tax reporting units, if the legal entity must pay taxes as a result of establishing a place of business within the jurisdiction.

## **FAQs for Define Initial Configuration**

### **What happens if I don't use the Enterprise Structures Configurator to set up my enterprise structures?**

The Enterprise Structures Configurator is an interview-based tool that guides you through setting up divisions, legal entities, business units, and reference data sets. The tool also enables you to assign reference data sets to business units and locations. You can set up multiple configurations to perform what-if scenarios, and then print each configuration to compare the resulting enterprise structure. If you do not use the Enterprise Structures Configurator, then you must set up your enterprise structure using the individual tasks that correspond to each enterprise component. In addition, you will not be able to set up multiple configurations and compare different scenarios. It is recommended that you use the Enterprise Structures Configurator.

### **What's an ultimate holding company?**

The legal entity that represents the top level in your organization hierarchy, as defined by the legal name entered for the enterprise. This designation is used only to create an organization tree, with the ultimate holding company as the top level, divisions and country holding companies as the second level, and legal employers as the third level.

## **Define Enterprise for Fusion Accounting Hub**

### **Enterprise: Explained**

An enterprise consists of legal entities under common control and management.

#### **Enterprise Defined**

When implementing Oracle Fusion Applications you operate within the context of an enterprise that has already been created in the application for you. This is either a predefined enterprise or an enterprise that has been created in the application by a system administrator.

An enterprise organization captures the name of the deploying enterprise and the location of the headquarters. There is normally a single enterprise organization in a production environment. Multiple enterprises are defined

when the system is used to administer multiple customer companies, for example, multiple tenants, or when a customer chooses to set up additional enterprises for testing or development.

Oracle Fusion Applications offers capabilities for multiple tenants to share the same applications instance for some human resources processes. If you offer business process outsourcing services to a set of clients, each of those clients may be represented as an enterprise within an Oracle Fusion Application instance. To support this functionality, system owned reference data such as sequences, sets, and flexfields are also defined within an enterprise.

In Oracle Fusion Applications, an organization classified as an enterprise is defined before defining any other organizations in the HCM Common Organization Model. All other organizations are defined as belonging to an enterprise.

## **Managing Enterprise Information for Non-Oracle Fusion HCM Users: Explained**

The Manage Enterprise HCM Information task includes default settings for your enterprise such as the employment model, worker number generation, and so on. If you are not implementing Oracle Fusion Human Capital Management (HCM), then the only action you may need to perform using this task is to change the enterprise name, if necessary. The other settings are HCM-specific and are not relevant outside of Oracle Fusion HCM.

---

---

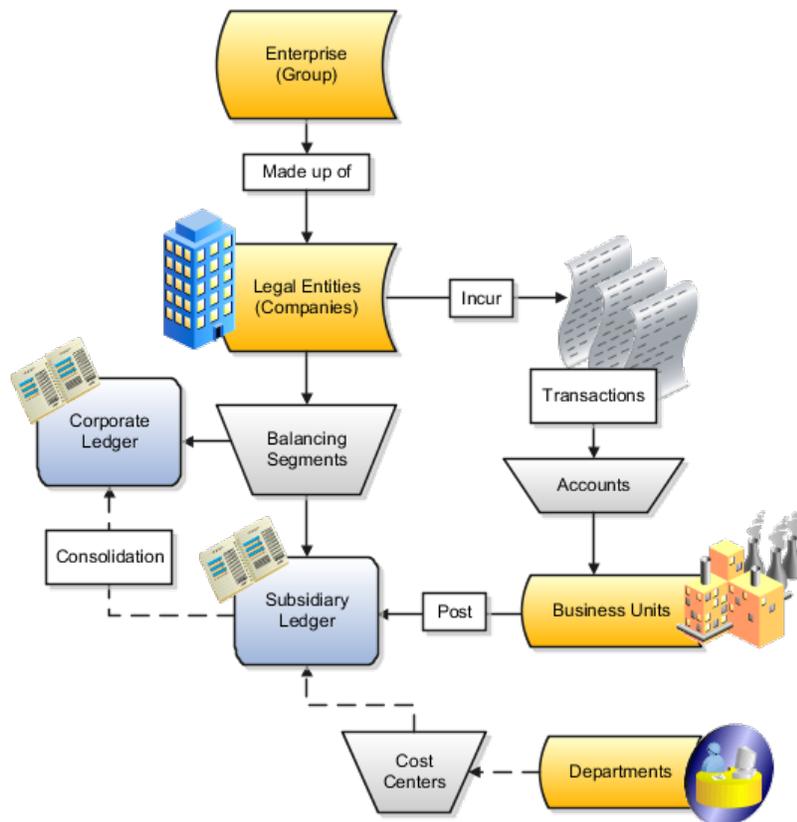
## Define Financial Reporting Structures

### Representing Your Enterprise Structure in Your Financial Reporting Structure: Overview

Represent your enterprise structures in your chart of accounts to track and report on your financial objectives and meet your reporting requirements. The benefit of representing your enterprise in the chart of accounts is the Oracle Fusion General Ledger functionality which includes multidimensional reporting with its Essbase tool. Segments included in the chart of accounts become dimensions in Essbase. The recorded data is automatically loaded into the Essbase cube when you post your journal entries. The Essbase tool includes powerful functionality for analysis and reporting on your financial data.

### Financial Enterprise Structure: How It Fits Together

Oracle Fusion Applications is an integrated suite of business applications that connects and automates the entire flow of the business process across both front and back office operations and addresses the needs of a global enterprise. The process of designing the enterprise structure, including the accounting configuration, is the starting point for an implementation. This process often includes determining financial, legal, and management reporting requirements and examining consolidation considerations.



## Accounting Configuration Components

The accounting configuration components are:

- **Ledgers:** A ledger is the main record-keeping structure. A ledger records transactional balances by defining a chart of accounts with a consistent calendar and currency, and accounting rules, implemented in an accounting method. The ledger is associated with the subledger transactions for the business units that are assigned to it, and provides context and accounting for them.
- **Balancing Segments:** Oracle Fusion Applications use the chart of accounts element, balancing segment, to represent and track both legal and management entities. Specifically, Oracle Fusion Applications provide a primary balancing segment to represent your legal entities, and additional balancing segments, you can implement for management reporting and analysis.

Balancing segments provide automatic balancing functionality by legal entity for journal entries, including intercompany and intracompany entries, suspense posting, and rounding imbalances.

- **Cost Centers:** Cost Centers aggregate elements of natural expenses to identify functional costs. A cost center can be the smallest segment of an organization for which costs are collected and reported. Not all cost centers represent organizations. A manager is assigned responsibility for cost control and is assigned both a department and a cost center; in which case the cost center and department might be identified with each other. However, a finance department manager might have separate cost centers for finance cost and audit costs, and an Research and Development

department manager might have separate cost centers for research and development.

Cost centers are represented by segment values in the chart of accounts that indicate the functional areas of your business, such as accounting, facilities, shipping, or human resources. You might keep track of functional areas at a detailed level, but produce summary reports that group cost centers into one or more departments. Cost center values are also used by Oracle Fusion Assets to assist the managers in tracking locations and accounting for assets assigned to their departments.

- **Accounts:** The account segment is a code in the chart of accounts that uniquely identifies each type of transactions recorded in the ledger and subledgers. The account segment is mapped to a dimension in the Essbase cube to enable reporting and inquiry. This functionality uses Oracle Fusion Business Intelligence Edition to analyze and drill into expense and revenue transactions.

## Representing Legal Entities, Business Units, Departments in Chart of Accounts

The following list provides information on how to represent legal entities, business units, and departments in chart of accounts.

- **Representing Legal Entities in the Chart of Accounts:** Legal entity is the term used in Oracle Fusion Applications for registered companies and other organizations recognized in law as having a legal existence and as parties with given rights and responsibilities.

Legal entities are created in the applications and then assigned balancing segment values, sometimes called company codes in your ledgers during accounting configuration.

- **Representing Business Units in the Chart of Accounts:** A business unit (BU) is part of an enterprise managed by a manager with profit and loss responsibility. The business unit is generally tracked in the chart of accounts. A business unit can be represented by a single ledger. For example, in countries where you need document sequencing for unique transaction sequencing within a legal entity, you can have a single ledger with a single business unit and legal entity.

A business unit can also be identified in the chart of accounts as a:

- Management segment value
- Balancing segment value
- Roll up of cost center segments using hierarchies

For example, a business unit manager is responsible for working capital utilization or overall asset utilization. You identify the business unit as a balancing segment value, to enable calculation of ratios for various utilization measurements.

A business unit is assigned to a primary ledger, as well as a default legal entity when it is configured. A BU identifies the primary ledger that subledger transactions are posted to, facilitating the use of more than one BU per general ledger. Each business unit posts transactions to a

single primary ledger. For example, a shared service center handles all the procurement functions for the entire company. The procurement transactions are posted to the business unit's ledger with intercompany entries to other ledgers as needed.

- **Representing Departments in the Chart of Accounts:** A department is an organizational structure with one or more operational objectives or responsibilities that exist independently of its manager and that has one or more employees assigned to it. The manager of a department is typically responsible for business deliverables, personnel resource management, competency management, and occasionally, for cost control and asset tracking.

In Oracle Fusion Applications, departments can be set up in Oracle Fusion Human Capital Management (HCM). If desired, they can also be represented by a unique segment in the chart of accounts or a group of cost center values.

## Modeling Your Financial Reporting Structure in Oracle Fusion: Example

This example uses a fictitious global company to demonstrate the analysis that can occur during the financial reporting structure planning process.

### Scenario

Your company, InFusion Corporation, is a multinational conglomerate that operates in the United States (US) and the United Kingdom (UK). InFusion has purchased an Oracle Fusion enterprise resource planning (ERP) solution including Oracle Fusion General Ledger and all of the Oracle Fusion subledgers. You are chairing a committee to discuss creation of a model for your global financial reporting structure including your chart of accounts for both your US and UK operations.

### InFusion Corporation

InFusion Corporation has 400 plus employees and revenue of \$120 million. Your product line includes all the components to build and maintain air quality monitoring (AQM) systems for homes and businesses. You have two distribution centers and three warehouses that share a common item master in the US and UK. Your financial services organization provides funding to your customers for the start up costs of these systems.

### Analysis

The following are elements you need to consider in creating your model for your financial reporting structure.

- Your company is required to report using US Generally Accepted Accounting Principles (GAAP) standards and UK Statements of Standard Accounting Practice and Financial Reporting Standards. How many ledgers do you need to achieve proper statutory reporting?
- Your financial services line of business has a different year end. Do you need a different calendar? Your financial services entity must report with average balance sheets. This feature of Oracle Fusion General Ledger provides you with the ability to track average and end-of-day balances, report average balance sheets, and create custom reports using both standard and average balances.
- Your corporate management requires reports showing total organizational performance with drill down capability to the supporting details. Do you need multiple balancing segment hierarchies to achieve proper rollup of balances for reporting requirements?

- Legal entity balancing options: Do you need to produce financial statements by one or more than one legal entity? Can you record multiple legal entities in one ledger or do you require multiple ledgers? Are you upgrading to Oracle Fusion Applications or a new install? If an upgrade, is your current financial reporting structure meeting your reporting needs?

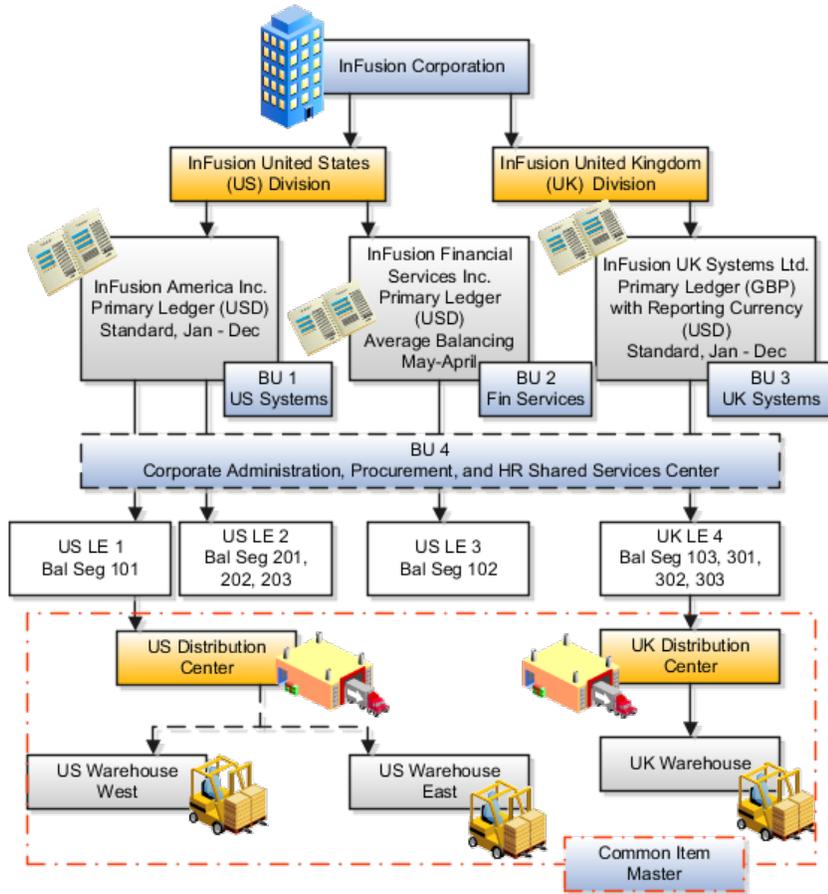
## Global Financial Reporting Model

The following figure and table summarize the model that your committee has designed and uses numerical values to provide a sample representation of your financial reporting structure. The model includes the following recommendations:

- Creation of three separate ledgers representing your separate legal entities:
  - InFusion America Inc.
  - InFusion Financial Services Inc.
  - InFusion UK Services Ltd.
- Data security is controlled by balancing segment values using Oracle Fusion General Ledger data access sets

Recommendations for the chart of accounts design include:

- Segments required for cost centers with hierarchical rollups to departments providing reporting at both the detail (cost center) and summary (department) level.
- Accounts configured to provide drill down capability to the subledger transactions, enabling analysis of data.



Decision	InFusion America, Inc.	InFusion Financial Services, Inc.	InFusion UK Systems, Ltd.
Type of Ledgers	Primary	Primary	Primary with the use of Reporting Currency functionality
Legal Entity Codes	US Legal Entity 1: US Corporate US Legal Entity 2: US Systems	US Legal Entity 3: US Financial Services	UK Legal Entity 4: UK Systems
Balancing Segments	101: US Corporate 201: US Systems Components 202: US Systems Installations 203: US Systems Maintenance	102: US Financial Services	103: UK Systems 301: Components 302: UK Systems Installations 303: UK Systems Maintenance
Currencies for Reporting	US Dollar (USD)	US Dollar (USD)	Great Britain Pounds Sterling (GBP) US Dollar (USD)
Calendar Ending date	December 31st	April 30th	December 31st

Business Units (BU)*	BU 1: US Systems BU 4: Corporate (Shared Service Center)	BU 2: Financial Services	BU 3: UK Systems
Balances Storage Method	Standard Balances	Average and Standard Balances	Standard Balances
Locations represented by cost centers in the chart of accounts.	Headquarters US Distribution Center (BU 1) US Warehouse West US Warehouse East	Headquarters	UK Distribution Center (BU 3) UK Warehouse

---

**Note**

In the chart of accounts, cost centers, with hierarchical rollups, represent business units. InFusion Corporation is also a legal entity but is not discussed in this example.

---

## Define Chart of Accounts

### Chart of Accounts: Explained

The chart of accounts is the underlying structure for organizing financial information and reporting. An entity records transactions with a set of codes representing balances by type, expenses by function, and other divisional or organizational codes that are important to its business.

A well-designed chart of accounts provides the following benefits:

- Effectively manages an organization's financial business
- Supports the audit and control of financial transactions
- Provides flexibility for management reporting and analysis
- Anticipates growth and maintenance needs as organizational changes occur
- Facilitates an efficient data processing flow
- Allows for delegation of responsibility for cost control, profit attainment, and asset utilization
- Measures performance against corporate objectives by your managers

The chart of accounts facilitates aggregating data from different operations, from within an operation, and from different business flows, thus enabling the organization to report using consistent definitions to their stakeholders in compliance with legislative and corporate reporting standards and aiding in management decisions.

Best practices include starting the design from external and management reporting requirements and making decisions about data storage in the general ledger, including thick versus thin general ledger concepts.

# Thick Versus Thin General Ledger: Critical Choices

Thick versus thin general ledger is standard terminology used to describe the amount of data populated and analysis performed in your general ledger. Thick and thin are the poles; most implementations are somewhere in between. Here are some variations to consider:

- A general ledger used in conjunction with an enterprise profitability management (EPM) product, which has data standardized from each operation, is designed as a thin general ledger. Use this variation if your solution is project based, and Oracle Fusion Projects is implemented. More detailed reporting can be obtained from the Projects system. In the thin general ledger, business units, divisions, and individual departments are not represented in the chart of accounts.
- A general ledger, with segments representing all aspects and capturing every detail of your business, with frequent posting, many values in each segment, and many segments, is called a thick general ledger. A thick general ledger is designed to serve as a repository of management data for a certain level of management. For example, a subsidiary's general ledger is designed to provide the upper management enough data to supervise operations, such as daily sales, without invoice details or inventory without part number details.
- A primary ledger and a secondary ledger, where one is a thick general ledger and the other a thin general ledger, provides dual representation for reporting requirements that require more than one ledger.

## Thin General Ledger

With a thin general ledger, you use the general ledger for internal control, statutory reporting, and tracking of asset ownership. You minimize the data stored in your general ledger. A thin general ledger has many of the following characteristics:

- Minimal chart of accounts
  - Short list of cost centers
  - Short list of natural accounts
    - Short list of cost accounts
    - Summary level asset and liability accounts
  - Low number of optional segments
- Infrequent posting schedule

A thin general ledger has natural accounts at a statutory reporting level, for example, payroll expense, rent, property taxes, and utilities. It has cost centers at the functional expense level, such as Research and Development (R&D) or Selling, General, and Administrative (SG&A) expense lines, rather than at department or analytic levels. It omits business unit, division, and product detail.

One example of an industry that frequently uses a thin general ledger is retail. In a retail organization, the general ledger tracks overall sales numbers by region. A retail point of sales product tracks sales and inventory by store, product, supplier, markup, and other retail sales measures.

### **Thick General Ledger**

With a thick general ledger, you use the general ledger as a detailed, analytic tool, performing analytic functions directly in the general ledger. Data is broken down by many reporting labels, and populated frequently from the subledgers.

You maximize the data stored in the general ledger. A thick general ledger has many of the following characteristics:

- Maximum use of the chart of accounts
  - Long list of natural accounts
  - Long list of cost centers
    - Long list of costing accounts
    - Detailed asset and liability accounts
- Frequent posting schedule

In a thick general ledger, you obtain detail for cost of goods sold and inventory balances and track property plant and equipment at a granular level. Cost centers represent functional expenses, but also roll up to departmental or other expense analysis levels. Using product and location codes in optional segments can provide reporting by line of business. Posting daily, at the individual transaction level, can maximize the data stored in the general ledger.

One example of an industry that frequently uses a thick general ledger is electronic manufacturers. Detail on the revenue line is tagged by sales channel. Product is structured differently to provide detail on the cost of goods sold line, including your bill of materials costs. The general ledger is used to compare and contrast both revenue and cost of goods sold for margin analysis.

### **Other Considerations**

Consider implementing a thick ledger if there are business requirements to do any of the following:

- Track entered currency balances at the level of an operational dimension or segment of your chart of accounts, such as by department or cost center
- Generate financial allocations at the level of an operational dimension or segment
- Report using multiple layered and versioned hierarchies of the operational dimension or segment from your general ledger

Consider implementing a thin ledger in addition to a thick ledger, if there are additional requirements for:

- Minimal disclosure to the authorities in addition to the requirements listed above. For example, in some European countries, fiscal authorities examine ledgers at the detailed account level.

- Fiscal only adjustments, allocations, and revaluations, which don't impact the thick general ledger.

The important consideration in determining if a thick ledger is the primary or secondary ledger is your reporting needs. Other considerations include how the values for an operational dimension or segment are derived and the amount of resources used in reconciling your different ledgers. If values for the operational dimension are always entered by the user like other segments of the accounting flexfield, then a thick primary ledger is the better choice.

However, if values for the operational dimension or segment are automatically derived from other attributes on the transactions in your subledger accounting rules, rather than entered in the user interface, then use a thick secondary ledger. This decision affects the amount of:

- Storage and maintenance needed for both the general ledger and subledger accounting entries
- System resources required to perform additional posting
- In summary, you have:
  - Minimum demand on storage, maintenance, and system resources with the use of a thin ledger
  - Greater demand on storage, maintenance, and system resources with the use of a thick ledger
  - Greatest demand on storage, maintenance and system resources with the use of both thick and thin ledgers

---

### Note

Generally speaking, there is a tradeoff between the volume of journals and balances created and maintained versus system resource demands. Actual performance depends on a wide range of factors including hardware and network considerations, transaction volume, and data retention policies.

---

### Summary

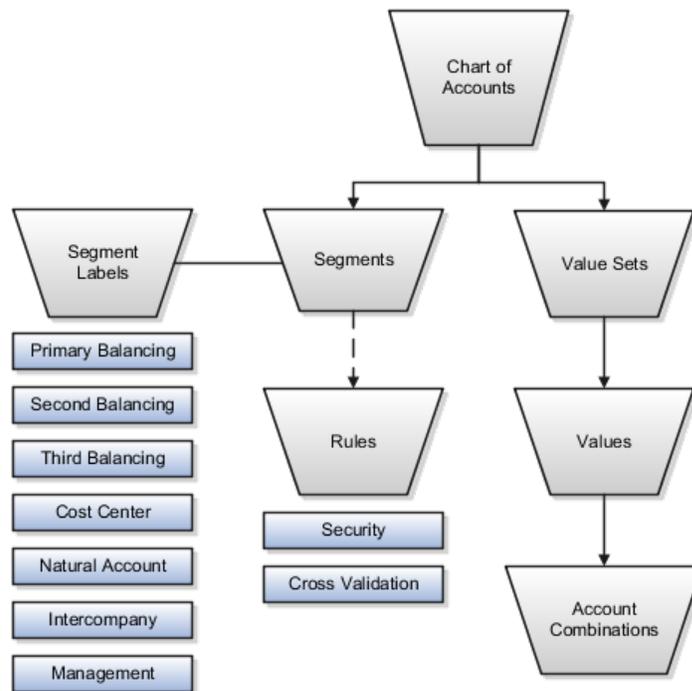
The factors you need to consider in your decision to use a thick or thin general ledger for your organization, are your:

- Downstream EPM system and its capabilities
- Business intelligence system and its capabilities
- Subledger systems and their capabilities and characteristics, including heterogeneity
- General ledger reporting systems and their capabilities
- Maintenance required for the thick or thin distributions and record keeping
- Maintenance required to update value sets for the chart of accounts segments
- Preferences of the product that serves as a source of truth

- Level at which to report profitability including gross margin analysis
- Industry and business complexity

## Chart of Accounts: How Its Components Fit Together

There are several important elements to the basic chart of accounts in Oracle Fusion Applications: a structure that defines the account values, segments, and their labels, and rules (security and validation). Account combinations link the values in the segments together and provide the accounting mechanism to capture financial transactions.



## Chart of Accounts

The chart of accounts defines the number and attributes of various segments, including the order of segments, the width of segments, prompts, and segment labels, such as balancing, natural account, and cost center.

The chart of accounts further defines the combination of value sets associated with each segment of the chart of accounts, as well as the type, default value, additional conditions designating the source of the values using database tables, and the required and displayed properties for the segments.

### Segments

A chart of accounts segment is a component of the account combination. Each segment has a value set attached to it to provide formatting and validation of the set of values used with that segment. The combination of segments creates the account combination used for recording and reporting financial transactions. Examples of segments that may be found in a chart of accounts are company,

cost center, department, division, region, account, product, program, and location.

### **Value Sets and Values**

The value sets define the attributes and values associated with a segment of the chart of accounts. You can think of a value set as a container for your values. You can set up your flexfield so that it automatically validates the segment values that you enter against a table of valid values. If you enter an invalid segment value, a list of valid values appears automatically so that you can select a valid value. You can assign a single value set to more than one segment, and you can share value sets across different flexfields.

### **Segment Labels**

Segment labels identify certain segments in your chart of accounts and assign special functionality to those segments. Segment labels were referred to as flexfield qualifiers in Oracle E-Business Suite. Here are the segment labels that are available to use with the chart of accounts.

- **Balancing:** Ensures that all journals balance for each balancing segment value or combination of multiple balancing segment values to use in trial balance reporting. There are three balancing segment labels: primary, second, and third balancing. The primary balancing segment label is required.
- **Cost Center:** Facilitates grouping of natural accounts by functional cost types, accommodating tracking of specific business expenses across natural accounts. As cost centers combine expenses and headcount data into costs, they are useful for detailed analysis and reporting. Cost centers are optional, but required if you are accounting for depreciation, additions, and other transactions in Oracle Fusion Assets, and for storing expense approval limits in Oracle Fusion Expense Management.
- **Natural Account:** Determines the account type (asset, liability, expense, revenue, or equity) and other information specific to the segment value. The natural account segment label is required.
- **Management:** Optionally, denotes the segment that has management responsibility, such as the department, cost center, or line of business. Also can be attached to the same segment as one of the balancing segments to make legal entity reporting more granular.
- **Intercompany:** Optionally, assigns the segment to be used in intercompany balancing functionality.

---

#### **Note**

All segments have a segment qualifier that enables posting for each value. The predefined setting is Yes to post.

---

### **Account Combinations**

An account combination is a completed code of segment values that uniquely identifies an account in the chart of accounts, for example 01-2900-500-123, might represent InFusion America (company)-Monitor Sales (division)-Revenue (account)-Air Filters (product).

## Rules

The chart of accounts uses two different types of rules to control functionality.

- Security rules: Prohibit certain users from accessing specific segment values. For example, you can create a security rule that grants a user access only to his or her department.
- Cross-validation rules: Control the account combinations that can be created during data entry. For example, you may decide that sales cost centers 600 to 699 should enter amounts only to product sales accounts 4000 to 4999.

## Chart of Accounts Structure and Instances: Critical Choices

In Oracle Fusion General Ledger, the chart of accounts model is framed around the concept of a chart of accounts structure, under which one or more chart of accounts structure instances can be created. There are critical choices that need to be considered when creating chart of accounts instances and structures.

### Chart of Accounts Structure

When creating the segments for the chart of accounts structure, you must enter segment sequence numbers that are consecutive and begin with the number one.

### Chart of Accounts Structure Instance

For segments in your chart of account structure instance that you expect to contain a large number of distinct values, you must perform the following steps:

- In the chart of accounts definition, mark the **segment query required** option as **selectively required**. In order to perform searches in the transactional user interface, you have to specify the following segment as a **mandatory search criteria**.
- Create required indexes in the `GL_CODE_COMBINATIONS` table for segments that are selectively required.

## Creating Chart of Accounts Structure and Instances: Examples

In Oracle Fusion General Ledger, the chart of accounts model is framed around the concept of a chart of accounts structure, under which one or more chart of accounts structure instances can be created. A chart of accounts structure defines the key attributes for your chart of accounts, such as the number of segments, the segment sequences, the segment names, segment prompts, segment labels, for example natural account and primary balancing, and default value sets.

The chart of accounts instance is exposed in the user interfaces and processes. By default, a chart of accounts instance inherits all the attributes of the chart of accounts structure, meaning that all instances of the same structure share a common shape and have the same segments in the same order. However, at the chart of accounts instance level, you can override the default value set assignments for your segments and assign a unique account hierarchy that determines the parent and child relationships between the value set values. At the chart of accounts instance level, determine if allow dynamic insertion is

enabled to generate new account combinations dynamically instead of creating them manually.

### Chart of Account Structure

You are creating a chart of accounts structure as you setup your chart of accounts for your enterprise, InFusion America, Inc. Follow these steps:

1. Navigate to the **Manage Chart of Accounts** page from the Functional Setup Manger by querying on **Manage Chart of Accounts** and clicking on the **Go To Task**.
2. Select **General Ledger** from the Module list of values and click **Search**.
3. Click **Manage Structures** to open the **Manage Key Flexfield Structures** page.
4. Select the **General Ledger** row and click the **Create** to open the **Create Key Flexfield Structure** page.
5. Enter a unique Structure Code, **INFUSION\_AM\_COA\_STRUCTURE**, and Name, **InFusion America COA Structure**. Provide an optional Description, **InFusion America Inc. Chart of Accounts Structure**.
6. Select the - Delimiter to visually separate your segment values.
7. Click **Save**.
8. To create a new segment, click the **Create** to open the **Create Key Flexfield Segment** page.
  - a. Enter the following parameters:

Parameter	Value
Segment Code	INFUSION_AM_CO
Name	InFusion America Company
Description	InFusion America Inc. Company
Sequence Number	1
Prompt	Company
Short Prompt	CO
Display Width	2
Column Name	Segment1
Default Value Set Code	INFUSION_AM_COMPANY

- b. Select a segment label, **Primary Balancing Segment**, to indicate its purpose within your chart of accounts.

---

#### Note

Two segment labels are required: primary balancing segment and natural account segment. These labels are not used with each other or with other labels in a specific segment.

- c. Click **Save and Close**.

- d. Click **Done**.
- e. Define additional segments following the same process.

### Chart of Account Instance

You are creating a chart of accounts instance as you setup your chart of accounts for your enterprise, InFusion America, Inc. Follow these steps:

1. Navigate to the **Manage Chart of Accounts** page from the Functional Setup Manger by querying on **Manage Chart of Accounts** and clicking on the **Go To Task**.
2. Select **General Ledger** from the Module list of values and click **Search**.
3. Select the **General Ledger** row and click **Manage Structure Instances** to open the **Manage Key Flexfield Structure Instance** page.
4. Click the **Create** icon to open the **Create Key Flexfield Structure Instance** page.
5. Enter a unique Structure Instance Code, **INFUSION\_AM\_COA\_INSTANCE**, and Name, **InFusion America COA Instance**. Provide an optional Description, **InFusion America Inc. Chart of Accounts Structure Instance**.
6. Select **Dynamic combination creation allowed** to indicate that you want to dynamically generate account combinations.
7. Associate your instance with your Structure Name, **InFusion America Structure**.

---

#### Note

By default, an instance inherits the key attributes of the associated structure. Some attributes, such as the value set assigned to each the segment, can be modified.

- 
8. Click **Save**.
  9. Optionally, select the segment row and click **Edit** to modify instance segments.
  10. Check **Required**, **Displayed**, and **Business intelligence enabled** check boxes.

---

#### Note

The Business Intelligence check box is only valid when enabled on segments with segment labels. Check the Required and Displayed options for all segments including those intended for future use. The recommended best practice is to define one segment for future use and set a default value. This ensures room for expansion in your chart of accounts and that the extra segment is populated in the account combinations.

- 
11. Click **OK**.
  12. Click **Save and Close**.
  13. Define additional instances following the same process.

---

**Note**

Alternatively, proceed directly with creating your value set values by selecting the corresponding **Value Set Code** in the Segment Instances table.

---

14. Click **Done**.
15. Click **Deploy Flexfield**.
16. Click **OK**.

## Creating One Chart of Accounts Structure with Many Instances: Example

In Oracle Fusion General Ledger, the chart of accounts model is framed around the concept of a chart of accounts structure, under which one or more chart of accounts structure instances can be created.

### Scenario

Your company, InFusion Corporation, is a multinational conglomerate that operates in the United States (US) and the United Kingdom (UK). InFusion has purchased an Oracle Fusion enterprise resource planning (ERP) solution including Oracle Fusion General Ledger and all of the Oracle Fusion subledgers. You are chairing a committee to discuss creation of a model for your global financial reporting structure including your charts of accounts for both your US and UK operations.

### InFusion Corporation

InFusion Corporation has 400 plus employees and revenue of \$120 million. Your product line includes all the components to build and maintain air quality monitoring (AQM) systems for homes and businesses.

### Analysis

In Oracle Fusion General Ledger, the chart of accounts model is framed around the concept of a chart of accounts structure, under which one or more chart of accounts structure instances can be created.

The chart of accounts structure provides the general outline of the chart of accounts and determines the number of segments, the type, the length, and the label (qualifier) of each segment. This forms the foundation of the chart of accounts definition object.

For each chart of accounts structure, it is possible to associate one or more chart of accounts structure instances. Chart of accounts structure instances under the same structure share a common configuration with the same segments, in the same order, and the same characteristics. Using one chart of accounts structure with multiple instances simplifies your accounting and reporting.

At the chart of accounts structure instance level, each segment is associated with a value set that conforms to the characteristic of that segment. For example, you assign a value set with the same segment type and length to each segment. You are using hierarchies with your chart of accounts segments. Each structure instance segment is assigned a tree code to indicate the source of the hierarchy

information for the associated value set. The same value set can be used multiple times within the same or across different chart of accounts instances within the same structure or in different structures. This functionality reduces your segment value creation and maintenance across your charts of accounts.

The collective assignment of value sets to each of the segments forms one chart of accounts instance. At the chart of accounts structure instance level, you can select to enable dynamic insertion. Dynamic insertion allows the creation of account code combinations automatically the first time your users enter that new account combination. The alternative is to create them manually. By deciding to enable dynamic insertion, you save data entry time and prevent delays caused by the manual creation of new code combinations. Well defined cross validation rules help prevent the creation of inappropriate account code combinations.

Perform deployment after a new chart of accounts structure and structure instances are defined or any of their modifiable attributes are updated. Deployment validates and regenerates the necessary objects to enable your charts of accounts and chart of accounts structure instances. By unifying and standardizing you organization's chart of accounts, you are positioned to take full advantage of future functionality in Oracle Fusion General Ledger.

In summary, you are recommending to your company to unify the organization's chart of accounts in a single chart of accounts structure based on chart of accounts commonalities across ledgers. You have also decided to use the chart of accounts structure instance construct to serve different accounting and reporting requirements by using value sets specific to each of your entities.

## Configuring Chart of Account Segment for Business Intelligence: Explained

To map the Oracle Fusion General Ledger Accounting Flexfield in Oracle Transaction Business Intelligence (BI) Repository file (RPD) for Oracle Fusion Financials, populate values in the Manage Key Flexfields user interface. These values enable the Chart of Accounts segments for Oracle Fusion Transactional BI and provide the mapping with BI Object names that are used as dimension for each of the Chart of Accounts segments.

Check each of the Chart of Accounts segments' **BI enabled** check box on all segments that you intend to map in the RPD by performing the following steps:

1. From your implementation project or the **Setup and Maintenance** page, query for **Manage Key Flexfields** and select the **Go to Task**.
2. Enter GL# in the **Key Flexfield Code** field.
3. Click **Search** button.
4. Click on **Manage Structure Instances** button.
5. Click the **Search** button.
6. Click on the desired chart of accounts and **Edit** icon.
7. Click on the desired segment and the **Edit** icon.
8. Edit each of the segments by checking the **BI enabled** check box.
9. Click on **Save** button. This should be done for all segments in every **Chart of Accounts Structure Instance** that you intend to be mapped in RPD.

10. Click the **Save and Close** button and the **Done** button.

Populate the **BI Object Name** for each of the **Segment Labels**. This name is the logical table name in the RPD which would be used as the dimension for the corresponding segment. Perform the following steps:

1. From your implementation project or the **Setup and Maintenance** page, query for **Manage Key Flexfields** and select the **Go to Task**.
2. Enter GL# in the **Key Flexfield Code** field.
3. Query for GL# as **Key Flexfield Code** in **Manage Key Flexfields** page.
4. Click **Search** button.
5. Chose **Actions** menu and click on **Manage Segment Labels**
6. Populate the **BI Object Name** for all the segment labels that are need to be mapped in the RPD.

Segment Label Code	BI Object Name
FA_COST_CTR	Dim - Cost Center
GL_BALANCING	Dim - Balancing Segment
GL_ACCOUNT	Dim - Natural Account Segment

7. Click the **Save** button.

---

#### Note

For all the non qualified segment labels, the **BI Object Name** should be populated with one of the following:

- Dim - GL Segment1
- Dim - GL Segment2
- Dim - GL Segment3
- Dim - GL Segment4
- Dim - GL Segment5
- Dim - GL Segment6
- Dim - GL Segment7
- Dim - GL Segment8
- Dim - GL Segment9
- Dim - GL Segment10

Deploy the flexfield using the **Deploy Flexfield** button from **Manage Key Flexfields** page.

## Cost Centers and Departments: Explained

A cost center represents the smallest segment of an organization for which costs are collected and reported. A department is an organization with one or more operational objectives or responsibilities that exist independently of its manager and has one or more workers assigned to it.

The following two components need to be considered in designing your enterprise structure:

- Cost centers
- Departments

### **Cost Centers**

A cost center also represents the destination or function of an expense as opposed to the nature of the expense which is represented by the natural account. For example, a sales cost center indicates that the expense goes to the sales department.

A cost center is generally attached to a single legal entity. To identify the cost centers within a chart of accounts structure use one of these two methods:

- Assign a cost center value in the value set for each cost center. For example, assign cost center values of PL04 and G3J1 to your manufacturing teams in the US and India. These unique cost center values allow easy aggregation of cost centers in hierarchies (trees) even if the cost centers are in different ledgers. However, this approach will require defining more cost center values.
- Assign a balancing segment value with a standardized cost center value to create a combination of segment values to represent the cost center. For example, assign the balancing segment values of 001 and 013 with cost center PL04 to represent your manufacturing teams in the US and India. This creates 001-PL04 and 013-PL04 as the cost center reporting values.

The cost center value of PL04 has a consistent meaning. This method requires fewer cost center values to be defined. However, it prevents construction of cost center hierarchies using trees where only cost center values are used to report results for a single legal entity. You must specify a balancing segment value in combination with the cost center values to report on a single legal entity.

### **Departments**

A department is an organization with one or more operational objectives or responsibilities that exist independently of its manager. For example, although the manager may change, the objectives do not change. Departments have one or more workers assigned to them.

A manager of a department is typically responsible for:

- Controlling costs within their budget
- Tracking assets used by their department
- Managing employees, their assignments, and compensation

---

#### **Note**

The manager of a sales department may also be responsible for meeting the revenue targets.

---

The financial performance of departments is generally tracked through one or more cost centers. In Oracle Fusion Applications, departments are defined

and classified as Department organizations. Oracle Fusion Human Capital Management (HCM) assigns workers to departments, and tracks the headcount at the departmental level.

The granularity of cost centers and their relationship to departments varies across implementations. Cost center and department configuration may be unrelated, identical, or consist of many cost centers tracking the costs of one department.

## FAQs for Manage Charts of Accounts

### How can I use future accounting segments?

To plan for future growth in the business organization that requires additional segments in the chart of accounts, extra segments can be added to the chart of accounts structure during your original implementation. Since all segments of the chart are required and have to be enabled, these unused segments can be assigned value sets that have a single value in the chart of accounts structure instance. This value is set as a default for that segment so that the extra segments are automatically populated when an account code combination is used.

## Manage Value Sets

### Value Sets: Explained

A value set is a set of valid values that you assign to a flexfield segment.

An end user enters a value into a flexfield segment while using the application. The flexfield validates the segment against the set of valid values that you configured as a value set and assigned to the segment.

For example, you can define a required format, such as a five digit number, or a list of valid values, such as green, red, and blue.

Flexfield segments are usually validated, and typically each segment in a given flexfield uses a different value set. You can assign a single value set to more than one segment, and you can share value sets among different flexfields.

---

#### Caution

Be sure changes to a shared value set are compatible with all flexfields segments using the value set.

---

Defining value sets involves making decisions about the following.

- Validation
- Security
- Precision and scale
- Usage and deployment

#### Validation

The following types of validation are available for value sets.

- Format only, where end users enter data rather than selecting values from a list

- Independent, a list of values consisting of valid values you specify
- Dependent, a list of values where a valid value derives from the independent value of another segment
- Subset, where the list of values is a subset of the values in an existing independent value set
- Table, where the values derive from a column in an application table and the list of values is limited by a WHERE clause

A segment that uses a format only value set does not present a list of valid values to users.

You can build a tree structure from the values in an independent value set whose data type is character.

---

### Note

Adding table validated value sets to the list of available value sets available for configuration is considered a custom task.

---

For more information, see the Oracle Fusion Applications Extensibility Guide.

### Security

Value set security only works in conjunction with usage within flexfield segments. If a value set is used standalone, meaning outside a flexfield, value set security is not applied, but Oracle Fusion data security is enforced.

You can specify that data security be applied to the values in flexfield segments that use a value set. Based on the roles provisioned to users, data security policies determine which values of the flexfield segment end users can view or modify.

Value set security applies at the value set level. If a value set is secured, every usage of it in any flexfield is secured. It is not possible to disable security for individual usages of the same value set.

Value set security applies to independent, dependent, or table-validated value sets.

Value set security applies mainly when data is being created or updated, and to key flexfield combinations tables for query purposes. Value set security does not determine which descriptive flexfield data is shown upon querying.

Security conditions defined on value sets always use table aliases. When filters are used, table aliases are always used by default. When predicates are defined for data security conditions, make sure that the predicates also use table aliases.

For key flexfields, the attributes in the view object that correspond to the code combination ID (CCID), structure instance number (SIN), and data set number (DSN) cannot be transient. They must exist in the database table. For key flexfields, the SIN segment is the discriminator attribute, and the CCID segment is the common attribute.

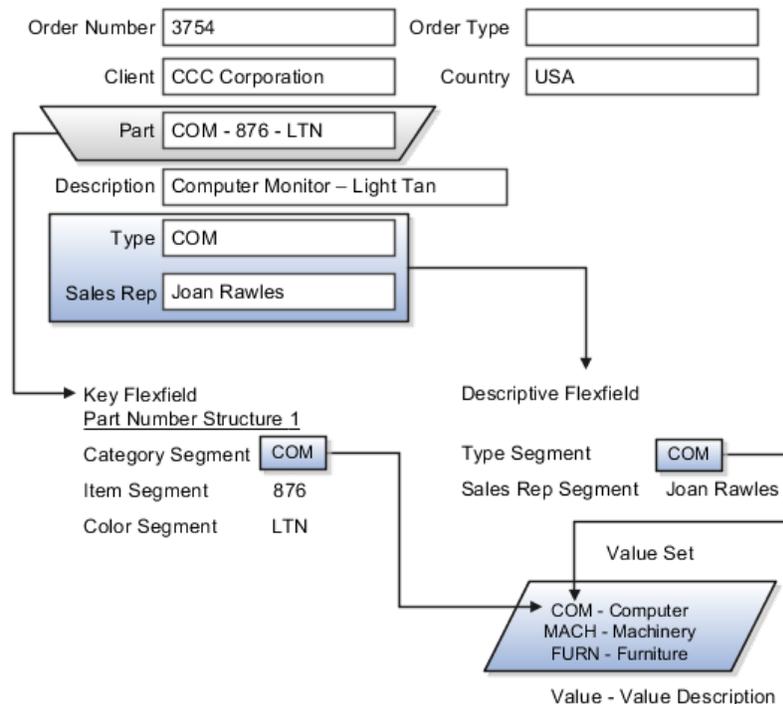
### Precision and Scale

For a value set with the data type Number, you can specify the precision (maximum number of digits user can enter) or scale (maximum number of digits following the decimal point).

## Usage and Deployment

The usage of a value set is the flexfields where that value set is used. The deployment status of flexfields in which the value set is used indicates the deployment status of the value set instance.

The figure shows a value set used by a segment in a key flexfield and the context segment of a descriptive flexfield.



For most value sets, when you enter values into a flexfield segment, you can enter only values that already exist in the value set assigned to that segment.

Global and context-sensitive segment require a value set. You can assign a value set to a descriptive flexfield context segment. If you specify only context values, not value sets for contexts, the set of valid values is equal to the set of context values.

## Chart of Accounts Values Sets: Critical Choices

A value set is the collection of account values that are associated with a segment of a chart of accounts structure instance. When creating values sets, consider the following critical choices:

- Module Designation
- Validation Type
- Format Assignments
- Security Rules
- Values Definition

## Module Designation

The module designation is used to tag value sets in Oracle Fusion Applications and sets the value sets apart during upgrades and other processes. Chart of accounts value sets upgraded from Oracle E-Business Suite Release 12 generically bear the module value of **Oracle Fusion Middleware**. When creating new value sets for a chart of accounts, the module can be specified as **Oracle Fusion General Ledger** to distinctly identify its intended use in an accounting flexfield, basically a chart of accounts.

## Validation Type

Assign one of the following validation types to chart of accounts value sets:

- **Independent:** The values are independently selected when filling out the segment in the account combination.
- **Table Validated:** The values are stored in an external table to facilitate maintenance and sharing of the reference data.

## Format Assignments

Value sets for chart of accounts must use the **Value Data Type** of **Character**. The **Value Subtype** is set to **Text**. These two settings support values that are both numbers and characters, which are typical in natural account segment values. Set the maximum length of the value set to correspond to the length of the chart of accounts segment to which it is assigned. Best practices recommend restricting values to **Upper Case Only** or **Numeric** values that are zero filled by default.

## Security Rules

If flexfield data security rules are to be applied to the chart of accounts segment associated with the value set, the **Enable Security** check box must be checked for the assigned value set. In addition, assign a data security resource name to enable creation of a data security object automatically for the value set. The data security object is used in the definition of flexfield data security rules.

## Value Definition

Once these basic characteristics are defined for the value set, values can be added to the set in the Manage Values page.

- Set the values to conform to the value set length and type.
- Enter the value, its description, and its attributes including the **Enable** check box, **Start Date**, and **End Date**.
- Assign the following attributes: **Parent** or **Summary** check box, **Posting is allowed**, and **Budgeting is allowed**.

---

## Note

If the value set is used with a natural account segment, the value also requires you set the **Natural Account Type**, with one of the following values: **Asset**, **Liability**, **Equity**, **Revenue** or **Expense**. Other attributes used are **Third Party Control Account**, **Reconciliation** indicator, and **Financial Category** used with Oracle Transaction Business Intelligence reporting.

Oracle Fusion General Ledger best practice is to define the values for the value set after the value set is assigned to a chart of accounts structure instance. Otherwise you are not able to define the mandatory value attributes, such as summary flag, posting allowed, and account type for natural account segment. The attributes must be added after the value set is assigned to a chart of accounts structure instance.

---

## Creating a Value Set for Your Chart of Accounts: Example

Create your value sets before creating your chart of accounts. A value set can be shared by different charts of accounts or across different segments of the same chart of accounts.

### Scenario

You are creating a company value set to be used in your chart of accounts for your enterprise, InFusion America, Inc. Follow these steps:

1. Navigate to the **Manage Chart of Accounts Value Sets** task from within your implementation project and click the **Go to Task**.
2. Click the **Create** icon on the toolbar of the Search Results table. The **Create Value Set** page opens.
3. Enter a unique Value Set Code, **InFusion America Company**, and an optional Description, **Company values for InFusion America Inc.**
4. Select **General Ledger** from the list in the Module field.
5. Select **Independent** as Validation Type.
6. Select **Character** as the Validation Data Type.
7. Click **Save and Close**.

## Manage Account Hierarchies

### Trees: Overview

Use the tree management feature in Oracle Fusion applications to organize data into hierarchies. A hierarchy contains organized data and enables the creation of groups and rollups of information that exist within an organization. Trees are hierarchical structures that enable several data management functions such as better access control, application of business rules at various levels of hierarchies, improved query performance, and so on.

For example, XYZ Corporation has two departments: Marketing and Finance. The Finance department has two functional divisions: Receivables and Payables. Defining a tree for the XYZ Corporation establishes a hierarchy between the organization and its departments, and between the departments and their respective functional divisions. Such a hierarchical modeling of organizational data could be used for executing several data management functions within that organization.

You can create one or more versions of trees, and they can be labeled for better accessibility and information retrieval. You can create trees for multiple data sources, which allow the trees to be shared across Oracle Fusion applications.

## Tree Structures

A tree structure is a representation of the data hierarchy, and guides the creation of a tree. A tree is an instance of the hierarchy as defined in the tree structure. Tree structures enable you to enforce business rules to which the data must adhere.

The root node is the topmost node of a tree. Child nodes report to the root node. Child nodes at the same level, which report to a common parent node, are called siblings. Leaves are details branching off from a node but not extending further down the tree hierarchy.

## Tree Versions

A tree is created having only one version. However, users can create more than one tree version depending on the need, and they can make changes to those versions. Depending on varying requirements, users can create one or more tree versions and publish all of them or some of them by making the versions active at the same time. Similar to any other version control system, versions of trees are maintained to keep track of all the changes that a tree undergoes in its life cycle.

## Tree Labels

Tree labels are short names associated with trees and tree structures and point directly to the data source. Tree labels are automatically assigned to the tree nodes. You can store labels in any table and register the label data source with the tree structure.

## Manage Tree Structures

### Tree Structures: Explained

A tree structure defines the hierarchy for creating trees and prescribes rules based on which trees are created, versioned, and accessed. You can associate multiple data sources with a tree structure. A tree is an instance of this hierarchy. Every tree structure can contain one or more trees.

You can create tree structures specific to an application but you can share tree structures across applications. If you apply version control to the tree structure, it is carried over to the trees that are based on the tree structure. Each tree version contains at least one root node. Occasionally, a tree version may have more than one root node.

An administrator controls the access to tree structures through a set of rules that are periodically audited for validity.

### Tree Structure Definition: Points to Consider

Defining a tree structure involves specifying several important pieces of information on the **Create Tree Structure: Specify Definition** page.

#### Tree Node Selection

The **Tree Node** table displays data in nodes that exist in the data hierarchy. You must select the correct and most appropriate tree node table to be able to define the tree structure, based on the tree hierarchy you want to establish. This

selection also affects the level of security that is set on a tree node and its child entities.

### **Tree Sharing Mode**

The following options are used to determine the mode of sharing a tree structure across the applications.

- Open: Indicates that the tree is associated with all reference data sets.
- Set ID: Indicates that the tree will be associated with a specific reference data set.

### **Creation Mode**

Indicates the source where the tree structure is being defined. For predefined tree structures select Oracle and for custom structures, select Customers.

### **Customization**

You can customize the predefined tree structures as well as the ones that you created. However, customizing the predefined tree structures involves certain level of access restrictions, and will be limited to specific tree nodes and downwards in hierarchy.

### **Multiple Tree Versions**

One or more trees and tree versions can be based on a tree structure. A tree structure can have one or more trees and tree versions based on it. Usually, only one active version is permitted at any given point of time. However, depending on the requirement, you can allow two or more tree versions to be in the active state for the same date range. This flexibility allows you to choose the tree version that you want to implement.

## **Managing Tree Structures: Points to Consider**

You can create, edit, and delete tree structures depending upon the requirement. You can also audit and change the status a tree structure.

### **Creating and Editing Tree Structures**

You can create trees on the basis of a tree structure. When you edit an active tree structure, the status of the tree structure and all associated trees and their versions change to draft. To reuse a tree structure, you can create a copy of it without copying the associated trees and tree versions. If you delete a tree structure, all the associated trees and tree versions are automatically deleted.

---

#### **Note**

For specific information on working with the predefined tree structures that exist in an Oracle Fusion application, refer to the specific product documentation.

---

### **Setting Status**

If you change the status of a tree structure, the status of the trees and tree versions associated with that tree structure also changes.

The following table lists the different statuses of a tree structure.

Status	Meaning
Draft	Yet to be published or is in a modified state.
Active	In use and based on which one or more trees or tree versions are created.
Inactive	Not in use.

## Tree Structure Audit Results: Explained

Use the tree structure audit results to verify the tree structure's correctness and data integrity. The audit results include the following details:

- The name of the validator, which is a specific validation check
- The result of the validation, including a detailed message
- Corrective actions to take if there are any validation errors

### Running an Audit

Setting the status of a tree structure to active automatically triggers an audit of that tree structure. You can also manually trigger an audit on the manage Tree Structures page, using **Actions - Audit**. The Tree Structure Audit Result table shows a list of validations that ran against the selected tree structure.

### Validation Details

The following table lists the validators used in the audit process and describes what each validator checks for. It also lists possible causes for validation errors and suggests corrective actions.

Validator	Description (what is checked)	Possible Cause for Validation Failure	Suggested Corrective Action
Restrict By Set ID	On the Manage Tree Structures: Specify Data Sources page, if the <b>Set ID</b> check box is selected to enable the <b>Restrict Tree Node List of Values Based on</b> option for a tree structure, each of its data source view objects must have a reference data set attribute. This validation does not take place when the check box is not selected.	Even when the check box is selected, one or more of its data source view objects does not contain a reference data set attribute.	If reference data set restriction is required for this tree structure, include a reference data set attribute on all data sources. Otherwise, deselect the check box.
Row Flattened Table Name	On the Manage Tree Structures: Specify Performance Options page, a valid row flattened table must be specified for the tree structure. It can either be the standard row flattened table <code>FND_TREE_NODE_RF</code> or a custom table.	<ul style="list-style-type: none"> <li>• The specified table does not exist in the database.</li> <li>• The specified table does not contain the same columns as the <code>FND_TREE_NODE_RF</code> table.</li> </ul>	Correct the row flattened table definition.

<p>Available Label Data Sources</p>	<p>On the Manage Tree Structures: Specify Data Sources page, if a labeling scheme is specified for the tree structure by selecting a list item from the <b>Labeling Scheme</b> list box, the label data source view object specified for each data source must be accessible, and the primary keys must be valid. This restriction does not apply when you select <b>None</b> from the <b>Labeling Scheme</b> list box.</p>	<ul style="list-style-type: none"> <li>• Any of the specified label data source view objects do not exist.</li> <li>• Any of the specified label data source view objects do not have primary keys.</li> <li>• When a label data source view object is initially defined, the database registers the primary keys for the view object. If the view object is later modified such that its primary keys no longer match the primary keys that were registered earlier, this validation fails.</li> </ul>	<ul style="list-style-type: none"> <li>• Correct the specified label data source view object.</li> <li>• Correct the primary keys of the specified label data source view object.</li> <li>• Either correct the primary keys in the label data source view object to match the primary keys that were earlier registered in <code>FND_TS_DATA_SOURCE</code>, or correct the primary keys registered in that table to match the new view object definition.</li> </ul>
-------------------------------------	---	---	---

Available Data Sources	Each data source view object specified for the tree structure must be accessible, and all its primary key attributes must be valid.	<ul style="list-style-type: none"> <li>• Any of the specified data source view objects do not exist.</li> <li>• When a data source view object is initially defined, the database automatically registers the primary keys for the view object if the <b>Use non-defined primary key columns</b> check box on the Data Source dialog box is not selected. If the check box is selected, the database registers the primary keys specified explicitly by the user on the Add Data Source dialog box. If the registered primary keys contain any duplicates, this validation fails.</li> <li>• The <b>Use non defined primary key columns</b> check box is selected in a data source, but the list of specified primary key columns does not match the primary keys defined in the corresponding data source view object.</li> <li>• Any common attribute that exists in both the data source view object and the tree node view object is not of the same data type in both view objects.</li> </ul>	<ul style="list-style-type: none"> <li>• Correct the specified data source view object.</li> <li>• Correct the duplicate column in the registered primary keys.</li> <li>• Correct the primary keys of the specified data source view object.</li> <li>• Correct any mismatch in data types.</li> </ul>
------------------------	---	---	---

Column Flattened Table Name	On the Manage Tree Structures: Specify Performance Options page, a valid column flattened table must be specified for the tree structure. It can either be the standard row flattened table <code>FND_TREE_NODE_CF</code> or a custom table.	<ul style="list-style-type: none"> <li>The specified table does not exist in the database.</li> <li>The specified table does not contain the same columns as the <code>FND_TREE_NODE_CF</code> table.</li> </ul>	Correct the column flattened table definition.
Restrict by Date	On the Manage Tree Structures: Specify Data Sources page, if the <b>Date Range</b> check box is selected to enable the <b>Restrict Tree Node List of Values Based on</b> option for a tree structure, each of its data source view objects must have effective start date and effective end date attributes. This validation does not take place when the check box is not selected.	Even when the check box is selected, one or more of its data source view objects does not contain effective start date and effective end date attributes.	If the date restriction is required for this tree structure, include the effective start date and effective end date attributes on all data sources. Otherwise, deselect the check box.
Tree Node Table Name	On the Manage Tree Structures: Specify Definition page, a valid tree node table must be specified for the tree structure. It can either be the standard row flattened table <code>FND_TREE_NODE</code> or a custom table.	<ul style="list-style-type: none"> <li>No table is specified in the <b>Tree Node Table</b> field.</li> <li>The specified table does not exist in the database.</li> <li>The specified table does not contain the same columns as the <code>FND_TREE_NODE</code> table.</li> </ul>	Correct the tree node table definition.
Allow Node Level Security	If the <b>Allow Node Level Security</b> option is set to No for the tree structure, the same option cannot be set to Yes on any of its data sources. This is a database setting that is not visible on the Manage Tree Structures page.	The option is set to No for the tree structure but one or more associated data sources have that option set to Yes.	Correct the option setting in the tree structure and their data sources.

### Specifying Data Sources for Tree Structures: Points to Consider

The data sources provide the items for establishing hierarchy in a tree structure. In the tree management infrastructure, these data sources are Oracle Application

Development Framework (ADF) business components view objects, which are defined by application development.

### Labeling Schemes

Selecting a labeling scheme determines how the tree nodes are labeled. You may select a labeling scheme to assign at the data source level, at the parent node level, or keep it open for customer assignment. You may also choose not to have any labeling scheme. However, if you decide to use any of the labeling schemes, you may need to select the following additional options, to restrict the list of values that appear under the selected tree node.

- **Allow Ragged Nodes:** To include nodes that have no child nodes, and are shorter than the remaining nodes in the entire hierarchy.
- **Allow Skip Level Nodes:** To include nodes that are at the same level but have parent nodes at different levels.

### Restriction of Tree Node Values

You can decide the depth of the tree structure by selecting an appropriate value from the list. Keeping the depth limit open renders an infinite list of values. Using the following options, you can restrict the list of values that appear for selection under a specific tree node.

- **Date Range:** Specifies whether a selection of nodes should be restricted to the same date range as the tree version.
- **Allow Multiple Root Nodes:** Allows you to add multiple root nodes when creating a tree version.
- **Reference Data Set:** Specifies whether a selection of nodes should be restricted to the same set as the tree.

### Data Source Values and Parameters

Tree data sources have optional data source parameters with defined view criteria and associated bind variables. You can specify view criteria as a data source parameter when creating a tree structure, and edit the parameters when creating a tree. Multiple data sources can be associated with a tree structure and can have well-defined relationships among them.

---

#### Note

Parameter values customized at the tree level override the default values specified at the tree-structure level.

---

The data source parameters are applied to any tree version belonging to that data source, when performing node operations on the tree nodes. Data source parameters also provide an additional level of filtering for different tree structures. The tree structure definition supports three data source parameter types.

- **Bound Value:** Captures any fixed value, which is used as part of the view criteria condition.
- **Variable:** Captures and binds a dynamic value that is being used by the data source view object. This value is used by the WHERE condition of the data flow.
- **View Criteria:** Captures the view criteria name, which is applied to the data source view object.

You can also specify which of the data source parameters are mandatory while creating or editing the tree structure.

View objects from the ADF business components are used as data sources. To associate the view object with the tree structure, you can pick the code from ADF business component view objects and provide the fully qualified name of the view object, for example, oracle.apps.fnd.applcore.trees.model.view.FndLabelVO.

## Specifying Performance Options for a Tree Structure: Points to Consider

Tree structures are heavily loaded with data. As a tree management guideline, use the following settings to improve performance of data rendering and retrieval.

- Row Flattening
- Column Flattening
- Column Flattened Entity Objects
- ADF Business Component View Objects

### Row Flattening

Row flattening optimizes parent-child information for run-time performance by storing additional rows in a table for instantly finding all descendants of a parent without initiating a CONNECT BY query. Row flattening eliminates recursive queries, which allows operations to perform across an entire subtree more efficiently.

To store row flattened data for the specific tree structure, users can either use the central `FND_TREE_NODE_RF` table or they can register their own row flattened table. For example, in a table, if Corporation is the parent of Sales Division (Corporation-Sales Division), and Sales Division is the parent of Region (Sales Division-Region), a row-flattened table contains an additional row with Corporation directly being the parent of Region (Corporation-Region).

### Column Flattening

Column flattening optimizes parent-child information for run-time performance by storing an additional column in a table for all parents of a child.

To store column flattened data for the specific tree structure, users can either use the central `FND_TREE_NODE_CF` table or they can register their own column flattened table. For example, in a table, if Corporation is the parent of Sales Division (Corporation-Sales Division), and Sales Division is the parent of Region (Sales Division-Region), a flattened table in addition to these columns, contains three new columns: Region, Sales Division, and Corporation. Although positioned next to each other, the column Region functions at the lower level and Corporation at the higher level, retaining the data hierarchy.

### Column Flattened Entity Objects

In the absence of a column-flattened table, if you need to generate the business component view objects for your tree structure for the flattened table, use the tree management infrastructure to correctly provide the fully qualified name of the entity object for the column flattened table.

### ADF Business Component View Objects

View objects from the ADF business components can also be used as data sources, eliminating the need to create new types of data sources. This field is to store the fully qualified name for the business component view object generated by the tree management for business intelligence reporting and usage. The business component view object is a combination of the tree data source and column flattened entity. Using this option prevents data redundancy and promotes greater reuse of existing data, thereby improving the performance of the tree structure.

## Manage Tree Labels

### Tree Labels: Explained

Tree labels are tags that are stored on tree nodes. You can store labels in any table and register the label data source with the tree structure. When a labeling scheme is used for trees, the selected labels are stored in the tree label entity and each tree node contains a reference to a tree label in the labeling scheme.

The following table lists the three ways in which tree labels are assigned to the tree nodes.

Labeling Scheme	Description
Level	Labels that are automatically assigned based on the data source to which the tree node belongs. A level label points to a specific data source. For example, in a tree that reflects the organizational hierarchy of an enterprise, all division nodes appear on one level and all department nodes on another.
Group	Labels that you can arbitrarily assign to tree nodes.
Depth	Labels that are automatically assigned based on the depth of the tree node within the tree. No manual assignment is performed. <hr/> <b>Note</b> In an unbalanced hierarchy, a level may not be equal to depth.

## Manage Trees and Tree Versions

### Managing Trees and Tree Versions: Points to Consider

You can create and edit trees and tree versions depending upon the requirement. A tree can have one or more tree versions. Typically, when changes are made to an existing tree, a new version is created and published.

#### Creating and Editing Trees

Trees are created based on the structure defined in the tree structure. You can create trees, modify existing trees, and delete trees. If you want to copy an existing tree, you can duplicate it. However, only the tree is duplicated and not its versions.

Creating a tree involves specifying the tree definition and specifying the labels that are used on its nodes. If the selected tree structure has data sources and parameters defined for it, they appear on the page allowing you to edit the parameter values at the tree node level.

---

#### Note

Parameter values customized at the tree level will override the default values specified at the tree-structure level.

---

#### Creating and Editing Tree Versions

Tree versions are created at the time of creating trees. A tree must contain a version.

Editing an existing tree provides you the choice to update the existing version. You can also edit the existing version that lies nested under the tree in the search results.

When you edit a tree version bearing Active status, the status changes to Draft until the modifications are saved or cancelled.

## Tree Version Audit Results: Explained

Use the tree version audit results to verify the tree version's correctness and data integrity. The audit results include the following details:

- The name of the validator, which is a specific validation check
- The result of the validation, including a detailed message
- Corrective actions to take if there are any validation errors

## Running an Audit

An audit automatically runs whenever a tree version is set to active. You can also manually trigger an audit on the Manage Trees and Tree Versions page, using **Actions - Audit**. The Tree Version Audit Result table shows a list of validations that ran against the selected tree version.

### Validation Details

The following table lists the validators used in the audit process and describes what each validator checks for. It also lists possible causes for validation errors and suggests corrective actions.

Validator	Description (what is checked)	Possible Cause for Validation Failure	Suggested Corrective Action
Effective Date	The effective start and end dates of the tree version must be valid.	The effective end date is set to a value that is not greater than the effective start date.	Modify the effective start and end dates such that the effective start date is earlier than the effective end date.
Root Node	On the Manage Tree Structures: Specify Data Sources page, if the <b>Allow Multiple Root Nodes</b> check box for the <b>Restrict Tree Node List of Values Based on</b> option is not selected, and if the tree structure is not empty, the tree version must contain exactly one root node. This validation does not take place if the check box is selected.	Even if the check box is deselected, the tree version has multiple root nodes.	Modify the tree version such that there is exactly one root node.
Data Source Max Depth	For each data source in the tree structure, on the Data Source dialog box, if the data source is depth-limited, the data in the tree version must adhere to the specified depth limit. This validation does not apply to data sources for which the <b>Maximum Depth</b> field is set to <b>Unlimited</b> .	The tree version has data at a depth greater than the specified depth limit on one or more data sources.	Modify the tree version such that all nodes are at a depth that complies with the data source depth limit.
Duplicate Node	On the Data Source dialog box, if the <b>Allow Duplicates</b> check box is not selected, the tree version should not contain more than one node with the same primary key from the data source. If the check box is selected, duplicate nodes are permitted.	Even when the check box is deselected, the tree version contains duplicate nodes.	Remove any duplicate nodes from the tree version.

Available Node	All nodes in the tree version should be valid and available in the underlying data source.	<ul style="list-style-type: none"> <li>• A node in the tree version does not exist in the data source. Deleting data items from the data source without removing the corresponding nodes from the tree version can result in orphaned nodes in the tree version. For example, if you added node A into your tree version, and subsequently deleted node A from the data source without removing it from the tree version, the validation fails.</li> <li>• The tree version contains a tree reference node, which references another tree version that does not exist.</li> </ul>	Remove any orphaned nodes from the tree version. Update tree reference nodes so that they reference existing tree versions.
Node Relationship	All nodes must adhere to the relationships mandated by the data sources registered in the tree structure.	The tree structure has data sources arranged in a parent-child relationship, but the nodes in the tree do not adhere to the same parent-child relationship. For example, if the tree structure has a Project data source with a Task data source as its child, Task nodes should always be under Project nodes in the tree version. This validation fails if there are instances where a Project node is added as the child of a Task node.	Modify the tree version such that the nodes adhere to the same parent-child relationships as the data sources.

SetID Restricted Node	On the Manage Tree Structures: Specify Data sources page, if the <b>Set ID</b> check box is selected to enable the <b>Restrict Tree Node List of Values Based on</b> option for each tree node, the underlying node in the data source must belong to the same reference data set as the tree itself. This restriction does not apply when the check box is not selected.	Even when the check box is selected, the tree version has nodes whose data source values belong to a different reference data set than the tree.	Modify the tree version such that all nodes in the tree have data sources with reference data set matching that of the tree.
Label Enabled Node	On the Manage Tree Structures: Specify Data Sources page, if a labeling scheme is specified for the tree structure by selecting a list item from the <b>Labeling Scheme</b> list box, all nodes should have labels. This restriction does not apply when you select <b>None</b> from the <b>Labeling Scheme</b> list box.	The tree structure has a labeling scheme but the tree version has nodes without labels.	Assign a label to any node that does not have a label.

Date Restricted Node	<p>On the Manage Tree Structures: Specify Data Sources page, if the <b>Date Range</b> check box is selected to enable the <b>Restrict Tree Node List of Values Based on</b> option for a tree structure, each node in the underlying data source must have an effective date range same as the effective date range of the tree version. This restriction does not apply if the check box is not selected.</p>	<p>Even when the check box is selected, there are data source nodes that have a date range beyond the tree version's effective date range. For example, if the tree version is effective from Jan-01-2012 to Dec-31-2012, all nodes in the tree version must be effective from Jan-01-2012 to Dec-31-2012 at a minimum. It is acceptable for the nodes to be effective for a date range that extends partly beyond the tree version's effective date range (for example, the node data source value is effective from Dec-01-2011 to Mar-31-2013). It is not acceptable if the nodes are effective for none or only a part of the tree version's effective date range (for example, the node data source value are effective only from Jan-01-2012 to June-30-2012).</p>	<p>Ensure that all nodes in the tree version have effective date range for the effective date range for the tree version.</p>
Multiple Active Tree Version	<p>On the Manage Tree Structures: Specify Definition page, if the <b>Allow Multiple Active Tree Versions</b> check box is not selected for the tree structure, there should not be more than one active tree version under a tree at any time. This restriction does not apply if the check box is selected.</p>	<p>Even when the check box is not selected, there is more than one active tree version in the tree for the same date range.</p>	<p>Set no more than one tree version to Active within the same date range and set the others to inactive or draft status.</p>
Range Based Node	<p>On the Data Source dialog box, if the <b>Allow Range Children</b> check box is not selected, range-based nodes are not permitted from that data source. This restriction does not apply if the check box is selected.</p>	<p>Even when the check box is not selected, there are range-based nodes from a data source.</p>	<p>Ensure that any range nodes in your tree version are from a data source that allows range children.</p>

Terminal Node	On the Data Source dialog box, if the <b>Allow Use as Leaves</b> check box is not selected, values from that data source cannot be added as leaves (terminal nodes) to the tree version. This restriction does not apply if the check box is selected.	Even when the check box is not selected, values from a data source are added as leaf nodes (terminal nodes).	Modify the tree version such that all terminal nodes are from data sources for which this check box is selected.
Usage Limit	On the Data Source dialog box, if the <b>Use All Values</b> option is selected to set the <b>Usage Limit</b> for the data source, every value in the data source must appear as a node in the tree. This restriction does not apply if <b>None</b> option is selected.	Even if the <b>Use All Values</b> option is selected, there are values in the data source that are not in the tree version.	For each data source value that is not yet available, add nodes to the tree version.

## Trees and Data Sources: How They Work Together

Data sources form the foundation for tree management in Oracle Fusion Applications. Tree structures, trees, and tree versions establish direct and real-time connectivity with the data sources. Changes to the data sources immediately reflect on the **Manage Trees and Tree Versions** page and wherever the trees are being used.

### Metadata

Tree structures contain the metadata of the actual data that is used in Oracle Fusion Applications. Tree structures contain the core business logic that is manifested in trees and tree versions.

### Data Storage

Trees and tree versions are built upon the tree structures. They employ the business rules defined in the tree structures and allow an application to select and enable a subset of trees to fulfill a specific purpose in that application.

### Access Control

Source data is mapped to tree nodes at different levels in the database. Therefore, changes you make to the tree nodes affect the source data. Access control set on trees prevents unwanted data modifications in the database. Access control can be applied to the tree nodes or anywhere in the tree hierarchy.

## Adding Tree Nodes: Points to Consider

Tree nodes are points of data convergence that serve as the building blocks of a tree structure. Technically, the node may be stored either in a product-specific table or in an entity that has been established by tree management as the default

storage mechanism. However, since all data in Oracle Fusion Applications usually have a storage home, only user-created data needs to be stored in an entity.

Nodes are attached to tree versions. Whenever you create or edit a tree version, you need to specify its tree node.

### **Managing Tree Nodes**

You can create, modify, or delete tree nodes on the **Tree Version: Specify Nodes** page. To add a tree node, ensure that the tree structure with which the tree version is associated is mapped to a valid data source. You can also duplicate a tree node if the multiple root node feature is enabled.

### **Node Levels**

In most trees, all nodes at the same level represent the same kind of information. For example, in a tree that reflects the organizational hierarchy, all division nodes appear on one level and all department nodes on another. Similarly, in a tree that organizes a user's product catalog, the nodes representing individual products might appear on one level and the nodes representing product lines on the next higher level.

When levels are not used, the nodes in the tree have no real hierarchy or reporting structure but do form a logical summarization structure. Strictly enforced levels mean that the named levels describe each node's position in the tree. This is natural for most hierarchies. Loosely enforced levels mean that the nodes at the same visual level of indentation do not all represent the same kind of information, or nodes representing the same kind of information appear at multiple levels. With loosely enforced levels, users assign a level to each node individually. The level is not tied to a particular visual position.

### **Node Types**

A tree node has the following node types.

- **Single:** Indicates that the node is a value by itself.
- **Range:** Indicates that the node represents a range of values and possibly could have many children. For example, a tree node representing account numbers 10000 to 99999.
- **Referenced Tree:** Indicates that the tree node is actually another version for the tree based on the same tree structure, which is not physically stored in the same tree. For example, a geographic hierarchy for the United States can be referenced in a World geographic hierarchy.

## **Importing Segment Values and Hierarchies: Explained**

Use Import Segment Values and Hierarchies process to load segment values and hierarchies if you maintain your chart of accounts reference data outside Oracle Fusion applications. You can load your segment values and hierarchies by populating two tables: `GL_SEGMENT_VALUES_INTERFACE` table and `GL_SEGMENT_HIER_INTERFACE` table, and running the Import Segment Values and Hierarchies process.

---

**Note**

You can load data to interface tables using predefined templates and the Load Interface File for Import scheduled process, which are both part of the External Data Integration Services for Oracle Cloud feature. For other implementations, optionally use this feature only if you have SFTP configured for it.

---

**The GL\_SEGMENT\_VALUES\_INTERFACE and GL\_SEGMENT\_HIER\_INTERFACE tables**

You can use GL\_SEGMENT\_VALUES\_INTERFACE to load segment values and GL\_SEGMENT\_HIER\_INTERFACE to load segment value hierarchies to Oracle Fusion applications. You can find details of the columns of the interface table in Oracle Enterprise Repository (OER) for Oracle Fusion Applications.

**Assigning Values for Columns in the GL\_SEGMENT\_VALUES\_INTERFACE table**

You must enter values in all columns of the interface table that require values, which includes all of the not null columns, in order for the Import Segment Values and Hierarchies process to be successful. Enter values in the following required columns of the interface table:

Column Name	Value
STATUS_CODE	Enter the value NEW to indicate that you are bringing new segment value data.
VALUE_SET_CODE	Enter the value set code for the segment values.
VALUE	Enter the segment value.
SUMMARY_FLAG	Select N if the segment value is a child value or Y if the segment value is a parent value.
ENABLED_FLAG:	Select Y to enable the segment value. Enter N to disable the segment value.
ACCOUNT_TYPE:	Enter the natural account type if the segment value is for natural account segment. Valid values are: A for Assets, L for Liabilities, E for Expenses, O for Owner's Equities, and R for Revenues.
ALLOW_POSTING_FLAG	Select Y if posting is allowed for this segment value. Select N if posting is not allowed.
OBJECT_VERSION_NUMBER	Enter default value of 1.

You can enter values for the following optional columns:

Column Name	Value
START_DATE_ACTIVE	Enter the start date of the segment value
END_DATE_ACTIVE	Enter the end date of the segment value.
THIRD_PARTY_CTRL_ACCOUNT	Enter the third party control account value. Valid values are: CUSTOMER, SUPPLIER, R for Restrict Manual Journals, Y, and N.
FINANCIAL_CATEGORY	Enter a financial category value for Oracle Transactional Business Intelligence reporting. Valid values are values defined in the FINANCIAL_CATEGORY lookup type.

DESCRIPTION	There are different description columns for different languages. To see segment value description in a different language installation, you need to populate the segment description for that language too.
-------------	---

The following columns should be left as null as Import Segment Values and Hierarchies process uses them for internal processing or does not use them in the current release.

- CREATION\_DATE
- CREATED\_BY
- LAST\_UPDATE\_DATE
- LAST\_UPDATE\_LOGIN
- LAST\_UPDATED\_BY
- SEGMENT\_VALUE\_INTERFACE\_ID
- REQUEST\_ID
- LOAD\_REQUEST\_ID

#### Assigning Values for Columns in the GL\_SEGMENT\_HEIR\_INTERFACE table

You must enter values in all columns of the interface table that require values, which includes all of the not null columns, in order for the Import Segment Values and Hierarchies process to be successful. Enter values in the following required columns of the interface table:

Column Name	Value
STATUS_CODE	Enter the value NEW to indicate that you are bringing new hierarchy data.
VALUE_SET_CODE	Enter the value set code for the segment values.
TREE_CODE	Enter the hierarchy name (tree code).
TREE_VERSION NAME	Enter the hierarchy version name (tree version name).
TREE_VERSION_START_DATE_ACTIVE	Enter the date that the tree version is activated.
TREE_VERSION_END_DATE_ACTIVE	Enter the date that the tree version is inactivated.
VALUE	Enter the segment value.
PARENT_VALUE	Select N if the segment value is a child value or Y if the segment value is a parent value.
DEPTH	Enter the depth of the hierarchy which shows the many ancestors the segment value has in the hierarchy.
OBJECT_VERSION_NUMBER	Enter default value of 1.

The following columns should be left as null as Import Segment Values and Hierarchies process uses them for internal processing or does not use them in the current release.

- CREATION\_DATE
- CREATED\_BY
- LAST\_UPDATE\_DATE
- LAST\_UPDATE\_LOGIN
- LAST\_UPDATED\_BY
- SEGMENT\_VALUE\_INTERFACE\_ID
- REQUEST\_ID
- LOAD\_REQUEST\_ID

## Loading Data to the Segment Value and Hierarchies Interface Tables: Explained

Load the segment values and hierarchies to the interface table by using the following steps.

1. Load segment values and hierarchies to comma separated values (csv) files. You can use the sample csv file or xls file that's provided in Oracle Enterprise Repository (OER) for Oracle Fusion Applications as a reference.
2. Upload the comma separated values (csv) file to the secure FTP server.
3. Run the Load Interface File for Import process.
4. After the data is loaded to the interface table, you can run the Import Segment Values and Hierarchies process to load the segment values and hierarchies.

## Maintain Segment Value Attributes

### Segment Labels: Explained

Segment labels identify certain segments in your chart of accounts structure and assign special functionality to those segments. Segment labels were referred to as flexfield qualifiers in Oracle E-Business Suite (EBS). Best practice is to assign each segment label one time within the chart of accounts structure. Here are the segment labels that are available to use with the chart of accounts structure.

Caution: Validations are not performed when segment labels are assigned, so verify that all are assigned correctly before using your chart of accounts.

#### Balancing

Ensures that all journals balance for each balancing segment value or combination of multiple balancing segment values. You can secure access to your primary balancing segment values only with data access sets. The general ledger application automatically calculates and creates balancing lines as required in

journal entries. For example, recognizing an entity's receivable and the other entity's payable. There are three balancing segment labels: primary, second, and third balancing. The primary balancing segment label is required.

### **Cost Center**

Represents the smallest segment of an organization for which costs are collected and reported. Facilitates grouping of natural accounts by functional cost types, accommodating tracking of specific business expenses across natural accounts. As cost centers combine expenses and headcount data into costs, they are useful for detailed analysis and reporting. Cost centers are optional, but required if accounting for depreciation, additions, and other transactions in Oracle Fusion Assets, and for storing expense approval limits in Oracle Fusion iExpense.

### **Natural Account**

Determines the account type (asset, liability, expense, revenue, or equity), whether posting is allowed, and other information specific to the segment value. The natural account segment is mapped to the Financial Category dimension in the balances cube to enable ad hoc reporting and transactional dashboards. This functionality uses Oracle Fusion Business Intelligence Enterprise Edition to analyze and drill into expense and revenue transactions. The natural account segment label is required.

### **Intercompany**

Optionally, assigns the segment to be used in intercompany balancing functionality. You cannot use the primary balancing or natural account segments as the intercompany segment. It is recommended that you use this segment and assign the same values to both the primary balancing and intercompany value sets to enable clear visibility of the due to and due from relationships inherent in intercompany accounting across the entire organization. Consider using the same values set for the primary balancing and intercompany segments to save maintenance and ensure completeness.

### **Management**

Optionally, denotes the segment that has management responsibility, such as the department, cost center, or line of business. Can be any segment, except the primary balancing or natural account segments. By designating a segment of your chart of accounts to the management segment, you can secure access to the management segment values with data access sets. By providing segment values to represent the lowest level of your organization, you can roll up results by line of business or other management criteria.

Note: Available in a future release. Do not assign this segment to your chart of accounts in Oracle Fusion Version 1.

## **Segment Labels: Example**

For a chart of accounts, each segment can be qualified by a label to distinctly indicate its purpose. This designation is also used by the Oracle Fusion

General Ledger processes to determine the proper way to display and process transactions and balances that are recorded.

### Scenario

You are creating your chart of accounts with six segments. Oracle Fusion General Ledger permits selection of up to thirty segments for your chart of accounts. You must have a minimum of three required segments, as determined below by the number of required segment labels (qualifiers). Required segment labels are:

- **Primary Balancing Segment:** Main balancing segment typically used to represent the company dimension of the organization. The segment set with this label cannot be set with another label.
- **Cost Center Segment:** Subunits of the company, such as cost or profit centers, or departments. You are required to create this segment if you are implementing Oracle Fusion Assets.
- **Natural Account Segment:** Classification of transactions and balances according to distinct account types: asset, liability, equity, revenue, and expense accounts. The segment set with this label cannot be set with another label.

The following optional segment labels are available and you are implementing all except for the Management Segment:

- **Second Balancing Segment:** Used to balance transactions, as needed, by an additional dimension beyond the primary balancing segment.
- **Third Balancing Segment:** Used to balance transactions, as needed, by an additional dimension beyond the primary and second balancing segments.
- **Management Segment:** Used in a future release. For Oracle Fusion Version 1, do not enable this qualifier.
- **Intercompany Segment:** Used to track intercompany due to and due from balances by identifying the specific trading company. The intercompany qualified segment cannot be set with any of the three balancing segment qualifiers. The values in this segment's value set need to be the same as the primary balancing segment.

Segment labels can only be assigned once within your chart of accounts. The following table shows how you are assigning the segment labels in your chart of accounts.

Segment	Segment Label
Company	Primary Balancing Segment
Cost Center	Cost Center and Second Balancing Segment
Location	Third Balancing Segment
Account	Natural Account Segment
Product Line	
Intercompany	Intercompany Segment

---

### Note

Validations are not performed when segment labels are assigned, so verify that all are assigned correctly before using your chart of accounts.

---

For Oracle Transactional Business Intelligence reporting, all labeled or qualified segments of the chart of accounts are automatically maintained in the data that reporting is based on. For non-qualified segments, the granularity of information stored in these segments is summarized and thus, Oracle Transactional Business Intelligence would not be able to provide detailed reporting along by these segments. If it is important to maintain the ability to perform detailed reporting on such segments, create custom labels to qualify these segments.

For example, one of the segments of the chart of accounts is based on product line, and none of the predefined segment labels above are applicable. It is important for the organization to derive product line based Oracle Transactional Business Intelligence reports. Create a custom label called Product Line to use to qualify the Product Line segment of the chart of accounts.

## Balancing Segments: Explained

Balancing segments ensure that all journals balance for each balancing segment value or combination of multiple balancing segment values. You can secure access to your primary balancing segment values only with data access sets. The general ledger application automatically calculates and creates balancing lines as required in journal entries. For example, recognizing an entity's receivable and the other entity's payable. There are three balancing segment labels: primary, second, and third balancing. The primary balancing segment label is required.

By enabling multiple balancing segments for your chart of accounts, it is possible to produce financial statements for each unique combination of segment values across not only one, but two or even three qualified balancing segments. This ability provides you greater insights into your operations as it affords you visibility along the critical fiscal dimensions you use to plan, monitor, and measure your financial performance.

The following explains processes that use balancing segments.

- Intercompany balancing: Adds lines to unbalanced journals using intercompany rules.
- Opening first period of the new accounting year: Calculates retained earnings amounts at the level of granularity that totals revenue and expense account balances for multiple balancing segment value combinations. This applies to standard and average balances.
- Importing journals: Adds lines using the suspense account on unbalanced journals.
- Posting journals: Adds additional lines to unbalanced journals for the following enabled account types:
  - Suspense
  - Rounding
  - Net income

- Retained earnings
- Cumulative translation adjustments from replication of revaluation journals to reporting currencies and for multiple reporting currency account type specific conversion
- Posting prior period journals: Calculates any income statement impact and posts to the appropriate retained earnings account.
- Translating balances: Supports multiple balancing segments for the following accounts:
  - Retained earnings: Calculated translated retained earnings are post to the retained earnings accounts by balancing segment. Retained earnings represents the summing of the translated revenue and expense accounts across multiple balancing segment values.
  - Cumulative translation adjustment: Amounts posted by balancing segment to these accounts represents currency fluctuation differences between ranges of accounts which use different rate types. For example, period end rates are used for asset and liability accounts and historical rates for equity accounts.
- Revaluing Balances: Supports multiple balancing segments when calculating gain or loss accounts.
- Creating Opening Balances: Initializes reporting currency balances by converting from the total primary currency. Any difference in the reporting currency amounts is offset by populating retained earnings accounts.
- Closing year end: Supports multiple balancing segments when calculating the income statement offset and closing account in the closing journals.

## Multiple Balancing Segments: Points to Consider

Oracle Fusion General Ledger supports tracking financial results at a finer level of granularity than a single balancing segment. In addition to the required primary balancing segment for the chart of accounts, which is typically associated with the company dimension of a business organization, two additional segments of the chart of accounts can be optionally qualified as the second and third balancing segments respectively. Possible chart of accounts segments that can be tagged as these additional balancing segments include cost center or department, additional aspects of a business commonly used in measuring financial results.

There are several points to consider in using multiple balancing segments:

- Journal entry processing
- Implementation timing
- Change options
- Migration adjustments

## Journal Entry Processing

Multiple balancing segments ensure that account balances come from journal entries where the debits equal the credits, and thus, the financial reports are properly generated for each unique instance of account value combinations across the balancing segments. Consider this option carefully as it provides more granular reporting but requires more processing resources.

## Implementation Timing

When considering implementing the optional second and third balancing segments, keep in mind that these chart of accounts segment labels are set from the beginning of time and are actively used by your ledgers. This is important to ensure that balances are immediately maintained in accordance with the necessary balancing actions to produce consistent financial reporting for the desired business dimensions. Multiple balancing segment ledgers that are not maintained from the beginning of time require extensive manual balance adjustments to catch up and realign the balances in accordance with the multiple balancing segments.

---

### Note

Do not set a segment already qualified as a natural account or intercompany segment as any of the three balancing segments. Validations are not performed when segment labels are assigned, so verify that all are assigned correctly before using your chart of accounts.

---

## Change Options

Once a segment has been enabled and designated as a balancing segment, you must not change the segment. Do not disable the segment or remove the segment labels. These settings must be consistently maintained throughout the life of the chart of accounts to control the accuracy and integrity of the financial data.

## Migration Adjustments

For charts of accounts migrated from Oracle E-Business Suite to Oracle Fusion General Ledger that use a segment with the secondary balance tracking segment qualifier, steps must be taken to ensure the proper transition to the second and third balancing segments. The required adjustments are extensive.

For ledgers associated with a migrated chart of accounts, its balances must be adjusted manually to be consistent with the second and third balancing segments as though these segment labels have been in place since the beginning of entries for these ledgers. This requires recomputing and updating of the following processes to reflect the correct balancing for each unique combination of segment values across the additional second and third balancing segments.

- Intercompany balancing
- Suspense posting
- Rounding imbalance adjustments on posting

- Entered currency balancing
- Revaluation gains or losses
- Retained earnings calculations at the opening of each new fiscal year
- Cumulative translation adjustments during translation

---

**Note**

All previously translated balances must also be purged, and new translations run to properly account for translated retained earnings and cumulative translation adjustments with the correct level of balancing.

---

## Using Multiple Balancing Segments: Example

This simple example illustrates balancing along two balancing segments for a simple chart of accounts with three segments.

### Scenario

Your company has a chart of accounts with two balancing segments and three segments, qualified as follows:

- Company: Primary balancing segment
- Cost Center: Second balancing segment
- Account: Natural account segment

The following multiple company and cost center journal has been entered to transfer advertising and phone expense from Company 1, Cost Center A to Company 2, Cost Center B.

Account	Debit	Credit
Company 1-Cost Center A-Advertising Expense Account	600	
Company 2-Cost Center B-Advertising Expense Account		600
Company 1-Cost Center A-Phone Expense Account	800	
Company 2-Cost Center B-Phone Expense Account		800

During the posting process, the last four lines are created to balance the entry across the primary and second balancing segments, company and cost center.

Account	Debit	Credit
Company 1-Cost Center A-Advertising Expense Account	600	
Company 2-Cost Center B-Advertising Expense Account		600

Company 1-Cost Center A-Phone Expense Account	800	
Company 2-Cost Center B-Phone Expense Account		800
Company 1-Cost Center A-Balancing Account		600
Company 2-Cost Center B-Balancing Account	600	
Company 1-Cost Center A-Balancing Account		800
Company 2-Cost Center B-Balancing Account	800	

## Segment Value Inheritance: Examples

The Segment Value Inheritance process simplifies the maintenance of the chart of accounts. When the characteristics of values in the value sets are updated, such as changes in enabled status, effective date, posting allowed status, or natural account type, all previously created account code combinations that referenced such values are not automatically updated by these changes. The Segment Value Inheritance process allows you to run a controlled process to update such existing account code combinations. This process maintains and corrects the current attribute settings for those account code combinations that contain the account values that were changed.

For account code combinations where the present settings need to be retained and not impacted by account attribute changes, activate the flag to preserve the account code combination's attribute. Activating the flag prevents those account code combination's attributes from being update when the Segment Value Inheritance process is run.

### Scenario

For example, there are three inactive account code combinations that share a common inactive cost center value of 110.

Company-Cost Center-Account	Enabled
01-110-5210	No
04-110-4310 (Preserve Attributes flag enabled)	No
03-110-6810	No

Cost center 110 went from being disabled to enabled. When the Segment Value Inheritance process is run, the following shows the result on these three account code combinations.

Company-Cost Center-Account	Enabled
01-110-5210	Yes
04-110-4310 (Preserve Attributes flag enabled)	No

03-110-6810	Yes
-------------	-----

---

**Note**

Once you disable a segment value and you log out of the system and back in, all code combinations containing that segment no longer work, even if the account combination still shows enabled in the account combination page. Use the Segment Value Inheritance process to set the enable flag correctly on the affected account code combinations.

---

## FAQs for Maintain Segment Values Attributes

### How can I change segments in an existing chart of accounts structure?

The chart of accounts structure and chart of accounts structure instance are fundamental constructs in the Oracle Fusion General Ledger accounting setup and cannot be altered once they are in use. The number of segments, the segments' order, each segment's label, length, type, and value set assignment are not updateable. These components set the foundation upon which accounting data is recorded for ledgers that use them. Careful and thoughtful planning must precede all decisions pertaining to defining the chart of accounts.

## Deploy Flexfields

### Flexfield Deployment: Explained

To use a flexfield at runtime, the flexfield must have been deployed at least once. Deployment generates or refreshes the Application Development Framework (ADF) business component objects that render the flexfield in a user interface. Flexfields are deployed for the first time during the application provisioning process.

After you configure or change a flexfield, you must deploy it to make the latest definition available to end users. You can deploy a flexfield to a sandbox for testing or to the mainline for use.

#### Deployment Status

Every flexfield has a deployment status.

A flexfield can have the following deployment statuses.

Deployment Status	Meaning
Edited	The flexfield metadata definition has not been deployed yet. Updates of the metadata definition are not applied in the runtime environment yet.
Patched	The flexfield metadata definition has been modified through a patch or through a data migration action, but the flexfield has not yet been deployed so the updated definition is not reflected in the runtime environment.

Deployed to Sandbox	The current metadata for the flexfield is deployed in ADF artifacts and available as a flexfield-enabled sandbox. The status of the sandbox is managed by the Manage Sandboxes task available to the Administrator menu of the Setup and Maintenance work area.
Deployed	The current metadata for the flexfield is deployed in ADF artifacts and available to end users. There have not been any changes to the flexfield since it was last deployed in the mainline.
Error	The deployment attempt in the mainline failed.

---

### Note

Whenever a value set definition changes, the deployment status of a flexfield that uses that value set changes to edited. If the change results from a patch, the deployment status of the flexfield changes to patched.

---

### Initial Deployment Status of Flexfields

The Oracle Fusion Applications installation loads flexfield metadata into the database. This initial load sets the flexfield status to Edited to indicate that the flexfield has not been deployed yet. The application provisioning process during installation deploys the predefined flexfields of the provisioned applications, which sets their status to Deployed if no errors are encountered.

When accessing a provisioned application, deployed flexfields are ready to use. In some cases, flexfield availability at runtime requires setup, such as defining key flexfields.

### Metadata Validation

Use the Validate Metadata command to view possible metadata errors before attempting to deploy the flexfield. Metadata validation is the initial phase of the Deploy and Deploy to Sandbox commands. By successfully validating metadata before running the deployment commands, you can avoid failures in the metadata validation phase of a deployment attempt. Errors in the metadata validation phase of deployment cause the deployment attempt to abort. Metadata validation results do not affect the deployment status of a flexfield.

## Flexfield Deployment Status: How It Is Calculated

Flexfield deployment status indicates how the flexfield metadata definition in the Oracle Fusion Applications database relates to the Application Development Framework (ADF) business components generated into a Metadata Services (MDS) repository.

## Settings That Affect Flexfield Deployment Status

If you have made a change to a flexfield and expect a changed deployment status, be sure you have saved your changes. No settings affect flexfield deployment status.

## How Flexfield Deployment Status Is Calculated

If the flexfield definition has been edited through the Define Flexfields activity task flows, the status is Edited. The latest flexfield metadata definition in the Oracle Fusion application diverges from the latest deployed flexfield definition. Any change, including if a value set used in a flexfield changes, changes the deployment status to Edited. If a flexfield has never been deployed, its status is Edited.

---

### Note

When an application is provisioned, the provisioning framework attempts to deploy all flexfields in that application.

---

If you deploy the flexfield to a sandbox successfully, the status is Deployed to Sandbox. The latest flexfield metadata definition in the Oracle Fusion application matches the metadata definition that generated ADF business components in a sandbox MDS repository. Whether the sandbox is active or not does not affect the deployment status. If the flexfield was deployed to a sandbox and has not been edited or re-deployed to the mainline since then, the status remains Deployed to Sandbox independent of whether the sandbox is active, or who is viewing the status.

If you deploy the flexfield successfully, meaning to the mainline, the status is Deployed. The latest flexfield metadata definition in the Oracle Fusion application matches the metadata definition that generated ADF business components in a mainline MDS repository. Change notifications are sent when a flexfield is deployed successfully to the mainline.

If either type of deployment fails so the current flexfield definition is not deployed, the status is Error. The deployment error message gives details about the error. The latest flexfield metadata definition in the Oracle Fusion application likely diverges from the latest successfully deployed flexfield definition.

If the flexfield definition has been modified by a patch, the status is Patched. The latest flexfield metadata definition in the Oracle Fusion application diverges from the latest deployed flexfield definition. If the flexfield definition was Deployed before the patch and then a patch was applied, the status changes to Patched. If the flexfield definition was Edited before the patch and then a patch was applied, the status will remain at Edited to reflect that there are still changes (outside of the patch) that are not yet in effect.

When a deployment attempt fails and you can access the Deployment Error Message for details.

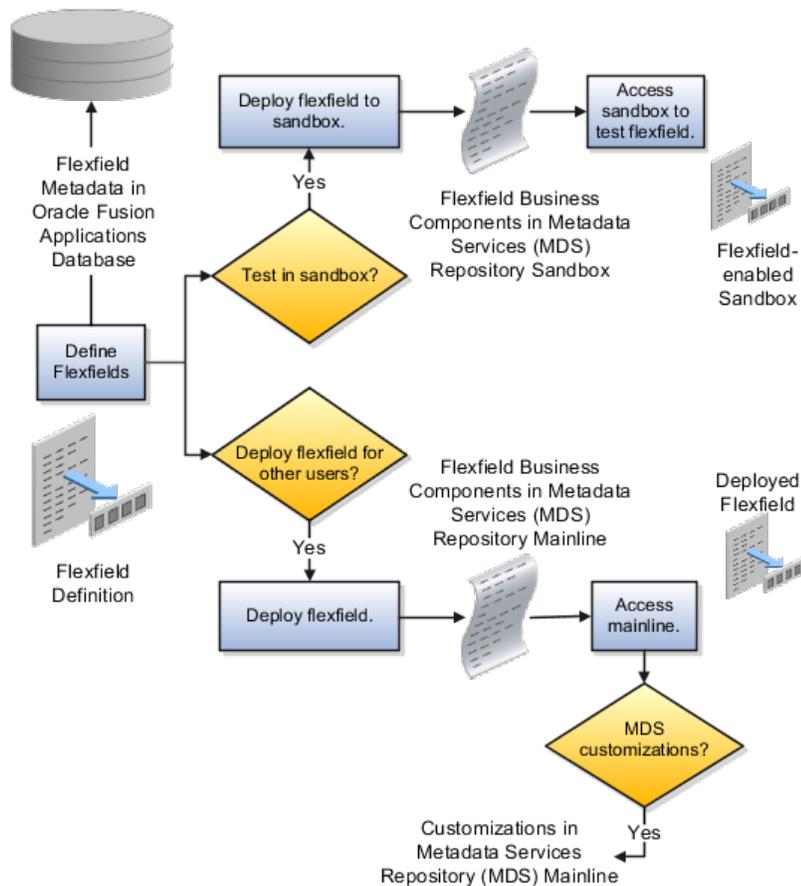
## Deploying a Flexfield-Enabled Sandbox: How It Works With Mainline Metadata

The flexfield definition in a sandbox corresponds to the flexfield metadata definition in the Oracle Fusion Applications database at the time the flexfield was deployed to the sandbox. When the flexfield is ready for end users, the flexfield must be deployed to the mainline.

A flexfield-enabled sandbox uses the following components.

- Flexfield metadata in the Oracle Fusion Applications database
- Flexfield business components in a sandbox Metadata Services (MDS) repository
- User interface customizations for the flexfield in the mainline MDS repository

The figure shows the two types of deployment available in the Manage Flexfield tasks of the Define Flexfields activity. Deploying a flexfield to a sandbox creates a sandbox MDS repository for the sole purpose of testing flexfield behavior. The sandbox is only accessible to the administrator who activates and accesses it, not to users generally. Deploying a flexfield to the mainline applies the flexfield definition to the mainline MDS repository where it is available to end users. After deploying the flexfield to the mainline, customize the page where the flexfield segments appear. Customization of the page in the sandbox MDS repository cannot be published to the mainline MDS repository.



### Sandbox Metadata Services Repository Data

Deploying the flexfield to a sandbox generates the Application Development Framework (ADF) business components of a flexfield in a sandbox MDS repository for testing in isolation.

---

#### Warning

Do not make changes to flexfield segment display features in a flexfield-enabled sandbox as these changes will be lost when deploying the flexfield to the mainline.

---

### **Mainline Metadata Services Repository Data**

The Oracle Fusion Applications database stores the single source of truth about a flexfield. From this the ADF business component objects that implement the flexfield in the runtime user interface are generated in the mainline MDS repository when the flexfield is deployed.

## **Deploying a Flexfield-Enabled Sandbox: Points to Consider**

Deploying a flexfield to a sandbox creates a flexfield-enabled sandbox . Each flexfield-enabled sandbox contains only one flexfield.

You can test the runtime behavior of a flexfield in the flexfield-enabled sandbox. If changes are needed, return to the Define Flexfield tasks to change the flexfield definition.

When you deploy a flexfield to sandbox, the process reads the metadata about the segments from the database, generates flexfield Application Development Framework (ADF) business component artifacts based on that definition, and stores in the sandbox only the generated artifacts derived from the definition.

### **Sandbox MDS Repository Data**

The sandbox data allows you to test the flexfield in isolation without first deploying it in the mainline where it could be accessed by users.

---

#### **Warning**

Do not make changes to flexfield segment display features in a flexfield-enabled sandbox as these changes will be lost when deploying the flexfield to the mainline.

---

### **Managing a Flexfield-Enabled Sandbox**

When you deploy a flexfield as a sandbox, that flexfield-enabled sandbox automatically gets activated in your user session. When you sign back in to see the changes, the sandbox is active in your session.

You can only deploy a flexfield to a sandbox using the Define Flexfields task flow pages.

You also can use the Manage Sandboxes feature in the Administration menu of the Setup and Maintenance work area to activate, access, or delete a flexfield-enabled sandbox.

---

#### **Note**

Whether you use the Define Flexfields or Manage Sandboxes task flows to access a flexfield-enabled sandbox, you must sign out and sign back in before you can see the changes you deployed in the runtime.

You cannot publish the flexfield from the sandbox to the mainline. You must use the Define Flexfields task flow pages to deploy the flexfield for access by users of the mainline because the flexfield configuration in the mainline is the single source of truth.

## Deploying Flexfields Using the Command Line: Explained

You can use the Manage Key Flexfields, Manage Descriptive Flexfields, and Manage Extensible Flexfields tasks to deploy flexfields. You can also use WebLogic Server Tool (WLST) commands for priming the Metadata Services (MDS) repository with predefined flexfield artifacts and for deploying flexfields.

The table describes the available commands.

WebLogic Server Tool Command	Description
<code>deployFlexForApp</code>	Deploys all flexfields for the specified enterprise application. Only flexfields whose status is other than deployed are affected by this command unless the option is enabled to force all flexfields to be deployed regardless of deployment status.  Initial application provisioning runs this command to prime the MDS repository with flexfield artifacts.
<code>deployFlex</code>	Deploy a single flexfield regardless of deployment status
<code>deployPatchedFlex</code>	Deploys flexfield changes that have been delivered using a flexfield Seed Data Framework (SDF) patch. Deploys flexfields that have a Patched deployment status.
<code>deleteFlexPatchingLabels</code>	Displays MDS label of flexfield changes for viewing and deleting patching labels.

Executing these commands outputs a report at the command line. The report provides the following information for every flexfield that is processed.

- Application identity (APPID)
- Flexfield code
- Deployment result, such as success or error

In case of errors, the report lists the usages for which the errors were encountered. If a runtime exception occurs, the output displays the traceback information. For each WLST flexfield command, adding the `reportFormat='xml'` argument returns the report as an XML string.

Consider the following aspects of command line deployment.

- Preparing to use the WLST flexfield commands

- Using the `deployFlexForApp` command
- Using the `deployFlex` command
- Using the `deployPatchedFlex` command
- Using the `deleteFlexPatchingLabels` command
- Exiting the WLST and checking the results

## Preparing To Use the WLST Flexfield Commands

You can only execute the WLST flexfield commands on a WebLogic Administration Server for a domain that has a running instance of the Oracle Fusion Middleware Extensions for Applications (Applications Core) Setup application.

For more information on deploying the Applications Core Setup application, see the Oracle Fusion Applications Developer's Guide.

Ensure that the AppMasterDB data source is registered as a JDBC data source with the WebLogic Administration Server and points to the same database as the ApplicationDB data source.

Start the WebLogic Server Tool (WLST) tool, if it is not currently running.

UNIX:

```
sh $JDEV_HOME/oracle_common/common/bin/wlst.sh
```

Windows:

```
wlst.cmd
```

Connect to the server, replacing the user name and password arguments with your WebLogic Server user name and password.

```
connect('wls_username', 'wls_password', 'wls_uri')
```

The values must be wrapped in single-quotes. The `wls_uri` value is typically `T3://localhost:7101`.

For more information on the WLST scripting tool, see the Oracle Fusion Middleware Oracle WebLogic Scripting Tool.

## Using the `deployFlexForApp` Command

The `deployFlexForApp` command translates the product application's predefined flexfield metadata into artifacts in the MDS repository.

---

### Important

This command is run automatically when you provision applications. However, after custom applications development, you must run the `deployFlexForApp` command after you configure your application to read the flexfield artifacts from the MDS repository and before you log into the application for the first time, even if there is no predefined flexfield metadata.

---

This command does not deploy flexfields that have a status of Deployed unless the force parameter is set to 'true' (the default setting is 'false').

For more information on priming the MDS partition with configured flexfield artifacts, see the Oracle Fusion Applications Developer's Guide.

From the WLST tool, execute the following commands to deploy the artifacts to the MDS partition, replacing `product_application_shortcode` with the application's short name wrapped in single-quotes.

```
deployFlexForApp('product_application_shortcode'[, 'enterprise_id']  
[, 'force'])
```

In a multi-tenant environment, replace `enterprise_id` with the Enterprise ID to which the flexfield is mapped. Otherwise, replace with 'None' or do not provide a second argument.

To deploy all flexfields regardless of their deployment status, set force to 'true' (the default setting is 'false'). If you want to deploy all flexfields in a single-tenant environment, you either can set `enterprise_id` to 'None', or you can use the following signature:

```
deployFlexForApp(applicationShortName='product_application_shortcode', force='true')
```

---

### Tip

The application's short name is the same as the application's module name.

---

For more information about working with application taxonomy, see the Oracle Fusion Applications Developer's Guide.

### Using the `deployFlex` Command

From the WLST tool, execute the following command to deploy a flexfield, replacing `flex_code` with the code that identifies the flexfield, and replacing `flex_type` with the flexfield's type, which is either DFF, KFF, or EFF.

```
deployFlex('flex_code', 'flex_type')
```

The values must be wrapped in single-quotes.

### Using the `deployPatchedFlex` Command

Use the `deployPatchedFlex` command for situations where the patching framework does not invoke the command, such as when an application has been patched offline.

If the installation is multi-tenant enabled, the command deploys all patched flexfields for all enterprises. This command is not intended to be invoked manually.

Check with your provisioning or patching team, or the task flows for managing flexfields, to verify that the flexfield has a Patched deployment status.

From the WLST tool, execute the following command to deploy the artifacts to the MDS partition of all flexfields that have a READY status.

```
deployPatchedFlex()
```

Execute the following command to deploy all flexfields that have either a READY status or an ERROR status.

```
deployPatchedFlex(mode='RETRY')
```

### Using the deleteFlexPatchingLabels Command

Whenever you deploy flexfield changes to MDS using the `deployPatchedFlex()` WLST command, an MDS label is created in the format `FlexPatchingWatermarkdate+time`. Use the `deleteFlexPatchingLabels` command to inquire about and delete these labels.

From the WLST tool, execute the `deployPatchedFlex()` command with no arguments to delete the flexfield patching labels.

To output a list of flexfield patching labels, execute the command with the `infoOnly` argument, as follows:

```
deleteFlexPatchingLabels(infoOnly='true')
```

### Exiting the WLST and Checking the Results

To exit the tool, execute the following command.

```
disconnect()
```

Optionally, sign into the application, access user interface pages that contain flexfields, and confirm the presence of flexfields for which configuration exists, such as value sets, segments, context, or structures.

## Manage Cross Validation Rules

### Cross Validation Rules: Overview

In Oracle Fusion General Ledger, use cross validation rules to determine the account combinations that are created dynamically as your users enter transactions or journal entries. Once enabled, a cross validation rule determines whether a selected value for a particular segment of the account combination can be combined with specific values in the other segments to form a new account combination.

A cross validation rule is defined in terms of a condition filter and a validation filter.

- The condition filter describes the event under which the rule will be evaluated. If the event specified in the condition filter is not applicable, then the rule will not be evaluated even if it is enabled.
- When the event specified in the condition filter is applicable, the validation filter condition must be satisfied before the account

combination can be created. The rule is evaluated using the following logic: If condition is satisfied, then perform specified validations.

For example, if your organization has determined that a certain company value, Operations, cannot use a specific cost center, Marketing, define the following cross validation rule to validate your accounts accordingly: If company is equal to Operations, then validate that cost center is not equal to Marketing.

If account combinations already exist and violate the newly enabled cross validation rules, these account combinations continue to be valid. Before disabling any existing account combinations that violate your rules and you are no longer using, move the balances in those accounts to the correct accounts. Then disable the account combinations manually to prevent further posting.

---

#### Note

Best practice is to define and enable cross validation rules before:

- Balances are loaded
  - Transactions or journal entries are imported or entered
  - Account combinations are created
- 

## Creating a Cross Validation Rule: Example

Create cross validation rules to prevent specific combinations of segment values in your account combinations, for example, preventing a particular cost center from being combined with a specific company value. Cross validation rules only affect the creation of new account combinations.

#### Scenario

Enter a new cross validation rule to prevent your InFusion America Inc. company value 01 from being combined with your marketing department value 300 in an account combination. Your company, InFusion America Inc. does not have a marketing department.

1. Navigate to the **Manage Cross-Validation Rules** task from within your implementation project, and then click the **Go to Task** icon.
2. Select your InFusion America chart of accounts. Click the **Create** icon.
3. Specify a unique rule Name, **IFAM01**, an optional Description, **Do not combine Marketing Department, 300 with InFusion America, company 01**.
4. Enter an optional effective From Date of today. Check **Enabled**.
5. Click on the **Change filter condition** on the Condition Filter. Enter Company equal to **01**. The cross validation rule evaluates if Company 01 was entered and if it was, then the validation process continues to evaluate the rule.

---

#### Note

If you do not specify any statement in the condition filter, then the rule is always evaluated.

- 
6. Click on the **Change filter condition** on the Validation Filter. Enter Cost Center equal to **300**. When the rule is evaluated, an account combination must contain a cost center other than 300 before it can be created.
  7. Enter an Error Message: **Cost Center 300 is not allowed with Company 01**. The message displays in the relevant user interfaces and processes when an account combination cannot be created because it violates the rule.
  8. Click **Save and Close**.

## Manage Chart of Accounts Mapping

### Mapping Chart of Accounts: Explained

The chart of accounts mapping feature supports the ability to correlate a source chart of accounts to a target chart of accounts to allow for the processing of balances or amounts. This is accomplished by either using segment rules, account rules, or a combination of both. A chart of accounts mapping is used by the posting process in propagating transactions from the primary ledger to its secondary ledger, providing the means to map the primary ledger chart of accounts to that of the secondary ledger. The mapping feature is used by both balance transfer processes for balance level secondary ledgers as well as cross ledger transfers, whereby balances from one ledger are copied to another ledger.

#### Segment Rules

Segment rules serve to map each segment of the target chart of accounts to an account value or segment in the source account. Three different mapping actions are available:

- Assign a constant value for a segment in the target chart of accounts
- Copy the value from the source segment to the corresponding target segment

---

#### Note

To use this action, the paired target and source segments must share identical values in their value sets.

- 
- Use roll up rules to aggregate source accounts to a corresponding target segment or account
    - Create a single value mapping when a specific detail source segment value is given a detail target segment value.
    - Use hierarchical roll up rules when a specific parent source value and all of its child segment values, are mapped to a given detail target

segment value. This provides the ability to process groups of source segment values in one single roll up rule.

- Define parent source values in roll up rules when date effective versions of the hierarchy are used with the accounting date of the transactions produced by the processes that reference the chart of accounts mapping. This gives the additional benefit of self maintaining mappings since the hierarchies referenced change with time, and the applicable child values are processed automatically.

## **Account Rules**

In addition to segment rules, define account rules for the chart of accounts mapping. Account rules map a complete target account code combination against one or more source account code combinations. The source account code combinations can be defined segment by segment using:

- Single detail account values
- Detail account value ranges
- Parent values for primary balancing and the natural account segments

---

### **Note**

When using parent values, its child values for the date effective version of the hierarchy, are processed when the mapping is called.

---

## **FAQ for Manage Chart of Accounts Mapping**

### **What's the difference between mapping with segment rules and mapping with account rules?**

Segment rules serve to map each segment of the target chart of accounts to an account value or segment in the source account of a secondary chart of accounts. A segment is only one part of the account code combination.

Account rules map a complete target account code combination against one or more source account code combinations.

---

### **Note**

Segment and account rules can be used alone or both types of mapping rules can be used in the same mapping set.

---

### **When do account rules override segment rules in the chart of accounts mapping?**

Segment rules and account rules can be exclusively used in a chart of accounts mapping, or you can use a combination of both. If there is an overlap between

the two types of rules, whereby a source account is mapped one way by the segment rules, and another by the account rules, the account rule supersedes. As such, segment rules can be used to more broadly define how to map the relationship between two charts of accounts on a segment by segment basis, and account rules can be used to more precisely delineate specific source account code combinations into their intended target accounts.

## Manage DRM Synchronization

### Integration with Data Relationship Management: Overview

Oracle Fusion Applications provides integration between Oracle Fusion Accounting Hub and Oracle Hyperion Data Relationship Management. The integration is included with Oracle Fusion Accounting Hub, but to use it, you must separately license Oracle Hyperion Data Relationship Management. This integration provides the ability to synchronize chart of accounts values and hierarchies. This integration can be used to establish corporate wide charts of accounts. You can use Hyperion Data Relationship Management, Fusion Edition, to store corporate charts of accounts values and hierarchies, and then update this information to both Oracle Fusion Accounting Hub and the Oracle E-Business Suite General ledger.

For more information on completing the post-installation setup for Data Relationship Management, see the Hyperion Data Relationship Management Integration with the Oracle Fusion and E-Business Suite General Ledgers white paper on My Oracle Support at <https://support.oracle.com>.

## Define Calendars

### Defining Accounting Calendars: Critical Choices

Define an accounting calendar to create your accounting year and the periods it contains. Specify common calendar options that the application uses to automatically generate a calendar with its periods. Specifying all the options makes defining a correct calendar easier and more intuitive with fewer errors. The choices you make when specifying the following options are critical, because it is difficult to change your accounting calendar after a period status is set to open or future enterable.

- Start Date
- Period Frequency
- Adjusting Period Frequency
- Period Name Format

---

#### Note

In Oracle Fusion, the common calendar types, monthly, weekly, 4-4-5, 4-5-4, 5-4-4, 4-week, quarterly, and yearly, are automatically generated. This

functionality makes it easier to create and maintain accounting calendars. By using the period frequency option, you no longer have to go through the tedious task of defining each period manually.

---

## Start Date

If you plan to run translation, specify a calendar start date that is a full year before the start date of the year of the first translation period for your ledger. Translation cannot be run in the first period of a calendar. Consider how many years of history you are going to load from your previous system and back up the start date for those years plus one more. You cannot add previous years once the first calendar period has been opened.

## Period Frequency

Use period frequency to set the interval for each subsequent period to occur, for example, monthly, quarterly, or yearly. If you select the period frequency of Other, by default, the application generates the period names, year, and quarter number. You specify the start and end dates. You must manually enter the period information. For example, select the period frequency of Other and enter 52 as the number of periods when you want to define a weekly calendar. For manually entered calendars, when you click the **Add Year** button, the application creates a blank year. Then, you must manually enter the periods for the new year. The online validation helps prevent erroneous entries.

---

## Note

In Oracle Fusion applications a calendar can only have one period frequency and period type. Therefore, if you have an existing calendar with more than one period type associated with it, during the upgrade from Oracle E-Business Suite, separate calendars are created based on each calendar name and period type combination.

---

## Adjusting Period Frequency

Use the adjusting period frequency to control when the application creates adjusting periods. For example, some of the frequencies you select add one adjusting period at year end, two at year end, or one at the end of each quarter. The default is None which adds no adjusting periods. If you select the frequency of Other, the **Number of Adjusting Periods** field is displayed. Enter the number of desired adjusting periods and then, manually define them.

## Period Name Format Region

The **User-Defined Prefix** field in the Period Name Format region is an optional feature that allows you to enter your own prefix. For example, define a weekly calendar and then enter a prefix of Week, - as the separator, and the period name format of Period numberYY fiscal year. The application creates the names of Week1-11, Week2-11, through Week52-11. The options for the **Format** field are predefined values. The list of values is filtered based on the selected separator and only displays the options that match the selected separator.

The year displayed in the period names is based on the selected period name format and the dates the period covers or if the period crosses years, on the year of the start date of the period. For example, April 10, 2010 to May 9, 2010 has the period name of Apr-10 and December 10, 2010 to January 9, 2011 has the name of Dec-10. If period frequency is Other, then the period format region is hidden. The application generates a temporary period name for calendars with period frequency of Other, using a fixed format of Period numberYY. You can override this format with your own customized period names.

---

**Note**

For an accounting calendar that is associated with a ledger, changing period names or adding a year updates the accounting period dimension in the balances cubes.

---

## Calendar Validation: How It Works with the Accounting Calendar

Calendar validation is automatic and prevents serious problems when you begin using the calendar. Once you set a calendar period status to open or future enterable, you cannot edit the period.

### Settings That Affect Calendar Validation

The calendar validation runs automatically when you save the calendar.

### How the Calendar Is Validated

The following table lists the validation checks performed when the accounting calendar is saved.

Validation Performed	Example of Data
Unique period number	2 assigned for two periods
Unique period name	Jan-11 entered twice
Period number beyond the maximum number of periods per year	13 for a 12 period calendar with no adjusting periods
Entered period name contains spaces	Jan 11
Single or double quotes in the period name	Jan '11
Nonadjusting periods with overlapping dates	01-Jan-2011 to 31-Jan-2011 and 30-Jan-2011 to 28-Feb-2011
Period date gaps	01-Jan-2011 to 28-Jan-2011 and 31-Jan-2011 to 28-Feb-2011
Missing period numbers	Periods 1 through 6 defined for a twelve month calendar
Period number gaps	1, 3, 5
Period numbers not in sequential order by date	Period 1 covers 01-Jan-2011 to 31-Jan-2011 and period 2 covers 01-Mar-2011 to 31-Mar-2011, and period 3 covers 01-Feb-2011 to 28-Feb-2011.

Quarter number gaps	1, 3, 4
Quarters not in sequential order by period	1, 3, 2, 4
Period start or end dates more than one year before or after the fiscal year	July 1, 2010 in a 2012 calendar

## FAQs for Manage Accounting Calendars

### How can I identify errors in my accounting calendar?

Oracle Fusion General Ledger identifies erroneous entries online as you enter a new calendar or change data on an existing calendar. The application also automatically validates the data when you save the calendar.

### What's the difference between calendar and fiscal period naming?

The period naming format determines the year that is appended to the prefix for each period in the calendar. For the example, your accounting year has a set of twelve accounting period with a start date of September 1, 2011 and the end date is August 31, 2012, with each period's date range following the natural calendar month date range.

Calendar period naming format: Select the calendar period format to append the period's start date's year to the prefix. For the period covering September 1, 2011 to December 31, 2011, then 2011 or just 11, depending on the period format selected, is appended to each period's name. For the remaining periods covering January 1, 2012 to August 31, 2012, then 2012 or 12, is appended to each period's name.

Fiscal period naming format: Select the fiscal period format to always append the period's year assignment to the prefix. If the accounting periods in the set of twelve are all assigned the year of 2012, then 2012 or just 12, depending on the period format selected, is appended to the period name of all 12 periods.

### When do I update an existing calendar?

Update an existing calendar before the new periods are needed as future periods, based on the future period setting in your accounting configuration. If a complete year has been defined and validated, use the **Add Year** button to add the next year quickly. Accept or change the new rows as required. For example, with the Other frequency type calendar, dates may differ from what the application generates.

### What happens if I upgrade my calendar from Oracle E-Business Suite Release 12?

The migration script assigns a period frequency that most closely matches your Oracle E-Business Suite Release 12 calendar. When you use the Oracle Fusion

applications Add Year functionality for the first time, you have an opportunity to review and change the period frequency. The Calendar Options page opens only for calendars upgraded from Release 12 to allow one time modification.

Make your changes to the period frequency, adjusting period frequency, and period name format, including the prefix and separator, as needed. Changes can not conflict with the existing upgraded calendar definition. Update the calendar name and description in the calendar header, as needed, for all calendars. Period details for a new year will be generated automatically based on the latest calendar options. You can also manually update the calendar. The modified calendar options affect future years only.

## Define Currencies

### Defining Currencies: Points to Consider

When creating or editing currencies, consider these points relevant to entering the currency code, date range, or symbol for the currency.

#### Currency Codes

You cannot change a currency code after you enable the currency, even if you later disable that currency.

#### Date Ranges

Users can enter transactions denominated in the currency only for the dates within the specified range. If you do not enter a start date, then the currency is valid immediately. If you do not enter an end date, then the currency is valid indefinitely.

#### Symbols

Even if you enter a symbol for a currency, the symbol is not always displayed when an amount is displayed in this currency. Some applications use currency symbols when displaying amounts. Others, like Oracle Fusion General Ledger, do not.

### Euro Currency Derivation: Explained

Use the Derivation Type, Derivation Factor, and Derivation Effective Date fields to define the relationship between the official currency (Euro) of the European Monetary Union (EMU) and the national currencies of EMU member states. For each EMU currency, you define its Euro-to-EMU fixed conversion rate and the effective starting date.

---

#### Note

If you need to use a different currency code for Euro, you can disable the predefined Euro currency and create a new one.

---

### **Derivation Type**

The **Euro currency** derivation type is used only for the Euro, and the **Euro derived** derivation type identifies national currencies of EMU member states. All other currencies do not have derivation types.

### **Derivation Factor**

The derivation factor is the fixed conversion rate by which you multiply one Euro to derive the equivalent EMU currency amount. The Euro currency itself should not have a derivation factor.

### **Derivation Effective Date**

The derivation effective date is the date on which the relationship between the EMU currency and the Euro begins.

## **FAQs for Manage Currencies**

### **When do I create or enable currencies?**

Create currencies to use, for example for reporting purposes, if they are not already provided. All currencies from the International Organization for Standardization (ISO) 4217 standard are provided.

Enable any currency other than USD for use in Oracle Fusion Applications, for example for displaying monetary amounts, assigning to sets of books, entering transactions, and recording balances. Only USD is enabled by default.

### **What's the difference between precision, extended precision, and minimum accountable unit for a currency?**

Precision is the number of digits to the right of the decimal point used in regular currency transactions. Extended precision is the number of digits to the right of the decimal point used in calculations for this currency, and it must be greater than or equal to the standard precision. For example, USD would have 2 for precision because amounts are transacted as such, for example \$1.00. For calculations, for example adding USD amounts, you might want the application to be more precise than two decimal digits, and would enter an extended precision accordingly.

---

### **Note**

Some applications use extended precision. Others, such as Oracle Fusion General Ledger, do not.

---

Minimum accountable unit is the smallest denomination for the currency. For example, for USD that would be .01 for the cent. This unit does not necessarily correspond to the precision for all currencies.

## Manage Conversion Rate Types

### Creating Conversion Rate Types: Critical Choices

Maintain different conversion rates between currencies for the same period with the Oracle Fusion General Ledger conversion rate types functionality. Four predefined daily conversion rate types are seeded: Spot, Corporate, User, and Fixed, allowing you to use different rate types for different business needs. During journal entry, the conversion rate is provided automatically by the General Ledger based on the selected conversion rate type and currency, unless the rate type is user. For user rate types, you must enter the conversion rate. Define additional rate types as needed. Set your most frequently used rate type as the default. Conversion rate types cannot be deleted.

Assign conversion rate types to automatically populate the associated rate for your period average and period end rates for the ledger. For example, you can assign the predefined rate type **Spot** to populate your period average rates and the predefined rate type **Corporate** to populate your period end rates. Period average and period end rates are used in translation of account balances.

Conversion rate types are used to automatically assign a rate when you perform the following accounting functions:

- Convert foreign currency journal amounts to ledger currency equivalents
- Convert journal amounts from source ledgers to reporting currencies or secondary ledgers
- Run Revaluation or Translation processes

In creating new conversion rates, decide whether to do the following:

- Enforce inverse relationships
- Select pivot currencies
- Select contra currencies
- Enable cross rates and allow cross rate overrides
- Maintain cross rate rules

#### Enforce Inverse Relationships

Check the **Enforce Inverse Relationship** check box to specify whether or not to enforce the automatic calculation of inverse conversion rates when defining daily rates.

Action	Results
Checked	<p>When you enter a daily rate to convert currency A to currency B, General Ledger automatically calculates the inverse rate, currency B to A, and enters it in the adjacent column. If either rate is changed, the application automatically recalculates the other rate.</p> <p>You can update the application calculated inverse rate, but once you do, the related rate is updated. The check box enforces that the inverse relationship is maintained but does not prevent you from updating the rates.</p>
Unchecked	<p>General Ledger calculates the inverse rate but you can change the rate and update the daily rates table without the corresponding rate being updated.</p>

### Select Pivot Currencies

Select a pivot currency that is commonly used in your currency conversions. A pivot currency is the central currency that interacts with contra currencies. For example, you set up a daily rate between the US dollar (USD) and the Euro currency (EUR) and another between the USD and the Canadian dollar (CAD). USD is the pivot currency in creating a rate between EUR and CAD. EUR and CAD are the contra currencies. Select the pivot currency from the list of values which contains those currencies that are enabled, effective, and not a statistical (STAT) currency. The description of the pivot currency is populated automatically based on the currency definition.

If you want the application to create cross rates against a base currency, define the base currency as the pivot currency. Selected pivot currencies can be changed in the Rate Types page.

### Select Contra Currencies

Select currencies available on the list of values as contra currencies. The available currencies are those currencies which are enabled, effective, not STAT currency, and not the pivot currency selected earlier. The description of the contra currency is populated automatically based on the currency definition. Add or delete contra currencies in the Contra Currencies region of the Rate Types page.

### Enable Cross Rates and Allow Cross Rate Overrides

Check the **Enable Cross Rates** check box to calculate conversion rates based on defined currency rate relationships. General Ledger calculates cross rates based on your defined cross rate rules. Associate your cross rate rules with a conversion rate type, pivot currency, and contra currencies. Cross rates facilitate the creation of daily rates by automatically creating the rates between contra currencies based on their relationship to a pivot currency. If the **Enable Cross Rates** check box is changed to unchecked after entering contra currencies, the application stops calculating cross rates going forward for that particular rate type. All the earlier calculated cross rates for that rate type remain in the database unless you manually delete them.

For example, if you have daily rates defined for the pivot currency, USD to the contra currency, EUR, and USD to another contra currency, CAD, the application will automatically create the rates between EUR to CAD and CAD to EUR. This prevents the need to manually define the EUR to CAD and CAD to EUR rates.

Check the **Allow Cross Rates Override** check box to permit your users to override application generated cross rates. If you accept the default of unchecked, the application generated cross rates cannot be overridden

### **Maintain Cross Rate Rules**

Define or update your cross rate rules at any time by adding or removing contra currency assignments. Add a contra currency to a cross rate rule and run the Daily Rates Import and Calculation process to generate the new rates. If you remove a cross rate rule or a contra currency from a rule, any cross rates generated previously for that contra currency remain unless you manually delete them. Changes to the rule are not retroactive and will not affect previously stored cross rates. The Cross Rate process generates as many rates as possible and skips currencies where one component of the set is missing.

---

#### **Note**

With a defined web service that extracts daily currency conversion rates from external services, for example Reuters, currency conversion rates are automatically updated for the daily rates and all cross currency relationships.

---

## **Using Rate Types: Examples**

There are four seeded conversion rate types in Oracle Fusion applications:

- Spot
- Corporate
- User
- Fixed

### **Scenario**

You are the general ledger accountant for InFusion America Inc. You are entering a journal entry to capture three transactions that were transacted in three different foreign currencies:

- Canadian dollar (CAD): A very stable currency
- Mexican Peso (MXP): A fluctuating currency
- Hong Kong dollar (HKD): An infrequently used currency

You enter two lines with accounts and amounts for each foreign currency transaction. Based on your company procedures, you select the appropriate rate

type to populate the rate for **Corporate** and **Spot** rate types from your daily rates table. You manually enter the current rate for the **User** rate type.

Currency Selected	Rate Type Selected	Reason
CAD	Corporate	Entered a periodic type of transaction. Your company has established a daily rate to use for the entire month across divisions for all transactions in CAD. CAD is a stable currency that only fluctuates slightly over the month.
MXP	Spot	Entered a periodic type of transaction. Your company enters daily rates each day for MXP because this currency is unstable and fluctuates.
HKD	User	Entered a one time transaction. Your company does not maintain daily rates in HKD.

---

#### Note

Your company does not currently use the **Fixed** rate type. From January 1, 1999, the conversion rate of the French franc (FRF) against the euro currency (EUR) was set at a fixed rate of 1 EUR to 6.55957 FRF. Your French operations were started in 2007, so you maintain all your French business records in the EUR.

---

## FAQs for Manage Conversion Rate Types

### What's the difference between spot, corporate, user, and fixed rate types?

Spot, corporate, user, and fixed conversion rate types differ based on the fluctuations of your entered foreign currency and your company procedures for maintaining daily rates.

Rate Type	Usage
Spot	For currencies with fluctuating conversion rates or when exact currency conversion is needed.
Corporate	For establishment of a standard rate across your organization for a stable currency.
User	For infrequent entries where your daily rates for the entered foreign currency are not set up.
Fixed	For rates where the conversion is constant between two currencies.

If you have infrequent foreign currency transactions, the user rate type can simplify your currency maintenance while providing an accurate conversion rate on the date of the transaction.

## What's a statistical unit currency type?

The statistical unit currency type is used only for the Statistical (STAT) currency. The Statistical currency is used to record statistics such as the number of items bought and sold. Statistical balances can be used directly in financial reports, allocation formulas, and other calculations.

## Manage Daily Rates

### Entering Daily Rates Manually: Worked Example

You are required to enter the daily rates for currency conversion from Great Britain pounds sterling (GBP) to United States dollars (USD) each day for your company InFusion America Inc.

Oracle Application Development Framework (ADF) Desktop Integration is an Excel add-in that must be loaded onto each client. Because ADF Desktop Integration is an add-in to Microsoft Office products, you can use this feature only if they have Microsoft Excel 2007 or above, Internet Explorer 7 or above, and Microsoft Windows 7, XP Professional SP2, or Vista. Users must download the installation files from **Navigator - Tools - Download Desktop Integrator Installer**.

### Entering Daily Rates

1. Navigate to the Period Close work area.  
Use the Period Close work area to link to close processes and currency process.
2. Click the Manage Currency Rates link.  
Use the Currency Rates Manager page to create, edit, and review currency rate types, daily rates, and historical rates.
3. Click the Daily Rates tab.  
Use the Daily Rates tab to review and enter currency rates.
4. Click the Create in Spreadsheet button.  
Use the Create Daily Rates spreadsheet to enter daily rates in a template that you can save and reuse.
5. Click in the From Currency field. Select the GBP - Pound Sterling list item.
6. Click in the To Currency field. Select the USD - US Dollar list item.
7. Click in the Conversion Rate field. Select the Spot list item.
8. Click in the From Conversion field. Enter the desired information into the From Conversion field. Enter a valid value e.g. "8/1/2011".
9. Click in the To Conversion Date field. Enter the desired information into the To Conversion Date field. Enter a valid value e.g. "8/1/2011".

10. Click in the Conversion Rate field. Enter the desired information into the Conversion Rate field. Enter a valid value e.g. "1.33225".
11. Click the Submit button. Click the OK button twice.
12. Review the Record Status column to verify that all rows were loaded successfully.
13. Save the template to use to enter daily rates frequently. You can save the spreadsheet to either a local drive or a shared network drive.

## Updating Currency Rates: Worked Example

You are required to change today's daily rates that were already entered. The rates you are changing are for currency conversion from Great Britain pounds sterling (GBP) to United States dollars (USD) for your company InFusion America Inc.

Currency conversion rates were entered by an automatic load to the Daily Rates table. They can also be entered through a spreadsheet.

### Updating Currency Rates

1. Navigate to the Period Close work area.  
  
Use the Period Close work area to link to close processes and currency process.
2. Click the Manage Currency Rates link.  
  
Use the Currency Rates Manager page to create, edit, and review currency rate types, daily rates, and historical rates.
3. Click the Daily Rates tab.  
  
Use the Daily Rates tab to review and enter currency rates.
4. Click the From Currency list. Select the GBP - Pound Sterling list item.
5. Click the To Currency list. Select the USD - US Dollar list item.
6. Enter the dates for the daily rates that you are changing. Enter today's date.
7. Click the Rate Type list. Select the Spot list item.
8. Click the Search button.
9. Click in the Rate field. Enter the new rate of 1.7 in the Rate field.
10. Click in the Inverse Rate field. Enter the new inverse rate of 0.58822 in the Inverse Rate field.
11. Click the Save button.



---

---

## Define Ledgers

### Ledgers and Subledgers: Explained

Oracle Fusion Applications reflect the traditional segregation between the general ledger and associated subledgers. Detailed transactional information is captured in the subledgers and periodically imported and posted in summary or detail to the ledger.

A ledger determines the currency, chart of accounts, accounting calendar, ledger processing options, and accounting method for its associated subledgers. Each accounting setup requires a primary ledger and optionally, one or more secondary ledgers and reporting currencies. Reporting currencies are associated with either a primary or secondary ledger.

The number of ledgers and subledgers is unlimited and determined by your business structure and reporting requirements.

#### Single Ledger

If your subsidiaries all share the same ledger with the parent company or they share the same chart of accounts and calendar, and all reside on the same applications instance, you can consolidate financial results in Oracle Fusion General Ledger in a single ledger. Use Oracle Fusion Financial Reporting functionality to produce individual entity reports by balancing segments. General Ledger has three balancing segments that can be combined to provide detailed reporting for each legal entity and then rolled up to provide consolidated financial statements.

#### Multiple Ledgers

Accounting operations using multiple ledgers can include single or multiple applications instances. You need multiple ledgers if one of the following is true:

- You have companies that require different account structures to record information about transactions and balances. For example, one company may require a six-segment account, while another needs only a three-segment account structure.
- You have companies that use different accounting calendars. For example, although companies may share fiscal year calendars, your retail operations require a weekly calendar, and a monthly calendar is required for your corporate headquarters.

- You have companies that require different functional currencies. Consider the business activities and reporting requirements of each company. If you must present financial statements in another country and currency, consider the accounting principles to which you must adhere.

### **Subledgers**

Oracle Fusion Subledgers capture detailed transactional information, such as supplier invoices, customer payments, and asset acquisitions. Oracle Fusion Subledger Accounting is an open and flexible application that defines the accounting rules, generates detailed journal entries for these subledger transactions, and posts these entries to the general ledger with flexible summarization options to provide a clear audit trail.

## **Ledgers: Points to Consider**

Companies account for themselves in primary ledgers, and, if necessary, secondary ledgers and reporting currencies. Your transactions from your subledgers are posted to your primary ledgers and possibly, secondary ledgers or reporting currencies. Local and corporate compliance can be achieved through an optional secondary ledger, providing an alternate accounting method, or in some cases, a different chart of accounts. Your subsidiary's primary and secondary ledgers can both be maintained in your local currency, and you can convert your local currency to your parent's ledger currency to report your consolidated financial results using reporting currencies or translation.

### **Primary Ledgers**

A primary ledger is the main record-keeping ledger. Like any other ledger, a primary ledger records transactional balances by using a chart of accounts with a consistent calendar and currency, and accounting rules implemented in an accounting method. The primary ledger is closely associated with the subledger transactions and provides context and accounting for them.

To determine the number of primary ledgers, your enterprise structure analysis must begin with your financial, legal, and management reporting requirements. For example, if your company has separate subsidiaries in several countries worldwide, enable reporting for each country's legal authorities by creating multiple primary ledgers that represent each country with the local currency, chart of accounts, calendar, and accounting method. Use reporting currencies linked to your country specific primary ledgers to report to your parent company from your foreign subsidiaries. Other considerations, such as corporate year end, ownership percentages, and local government regulations and taxation, also affect the number of primary ledgers required.

### **Secondary Ledgers**

A secondary ledger is an optional ledger linked to a primary ledger for the purpose of tracking alternative accounting. A secondary ledger can differ from its primary ledger by using a different accounting method, chart of accounts, accounting calendar, currency, or processing options. All or some of the journal entries processed in the primary ledger are transferred to the secondary ledger, based on your configuration options. The transfers are completed based on the conversion level selected. There are four conversion levels:

- **Balance:** Only Oracle Fusion General Ledger balances are transferred to the secondary ledger.
- **Journal:** General Ledger journal posting process transfers the journal entries to the secondary ledger.
- **Subledger:** Oracle Fusion Subledger Accounting creates subledger journals to subledger level secondary ledgers as well as reporting currencies.
- **Adjustments Only:** Incomplete accounting representation that only holds adjustments. The adjustments can be manual or detailed adjustments from Subledger Accounting. This type of ledger must share the same chart of accounts, accounting calendar, and period type combination, and currency as the associated primary ledger.

---

**Note**

A full accounting representation of your primary ledger is maintained in any subledger level secondary ledger.

---

Secondary ledgers provide functional benefits, but produce large volumes of additional journal entry and balance data, resulting in additional performance and memory costs. When adding a secondary ledger, consider your needs for secondary ledgers or reporting currencies, and select the least costly data conversion level that meets your requirements. For secondary ledgers, the least costly level is the adjustment data conversion level because it produces the smallest amount of additional data. The balance data conversion level is also relatively inexpensive, depending upon how often the balances are transferred from the primary to the secondary ledger. The journal and subledger data conversion levels are much more expensive, requiring duplication of most general ledger and subledger journal entries, as well as general ledger balances.

For example, you maintain a secondary ledger for your International Financial Reporting Standards (IFRS) accounting requirements, while your primary ledger uses US Generally Accepted Accounting Principles (GAAP). You decided to select the subledger level for your IFRS secondary ledger. However, since most of the accounting is identical between US GAAP and IFRS, a better solution is to use the adjustment only level for your secondary ledger. The subledger level secondary ledger requires duplication of most subledger journal entries, general ledger journal entries, and general ledger balances. With the adjustment only level, your secondary ledger contains only the adjustment journal entries and balances necessary to convert your US GAAP accounting to the IFRS accounting, which uses a fraction of the resources that are required by full subledger level secondary ledger.

Following are scenarios that may require different combinations of primary and secondary ledgers:

- The primary and secondary ledgers use different charts of accounts to meet varying accounting standards or methods. A chart of accounts mapping is required to instruct the application how to propagate balances from the source (primary) chart of accounts to the target (secondary) chart of accounts.
- The primary and secondary ledgers use different accounting calendars to comply with separate industry and corporate standards.

---

**Note**

Use the same currency for primary and secondary ledgers to avoid difficult reconciliations, if you have the resources to support the extra posting time and data storage. Use reporting currencies or translations to generate the different currency views needed to comply with internal reporting needs and consolidations.

---

**Reporting Currencies**

Reporting currencies maintain and report accounting transactions in additional currencies. Each primary and secondary ledger is defined with a ledger currency that is used to record your business transactions and accounting data for that ledger. It is advisable to maintain the ledger in the currency in which the majority of its transactions are denominated. For example, create, record, and close a transaction in the same currency to save processing and reconciliation time. Compliance, such as paying local transaction taxes, is also easier using a local currency. Many countries require that your accounting records be kept in their national currency.

If you need to maintain and report accounting records in different currencies, you do this by defining one or more reporting currencies for the ledger. There are three conversion levels for reporting currencies:

- **Balance:** Only General Ledger balances are converted into the reporting currency using translation.
- **Journal:** General Ledger journal entries are converted to the reporting currency during posting.
- **Subledger:** Subledger Accounting creates subledger reporting currency journals along with primary ledger journals.

---

**Note**

A full accounting representation of your primary ledger is maintained in any subledger level reporting currency. Secondary ledgers cannot use subledger level reporting currencies.

Of the three data conversion levels available, the balance data conversion level is typically the least expensive, requiring duplication of only the balance level information. The journal and subledger data conversion levels are more expensive, requiring duplication of most general ledger and subledger journal entries, as well as general ledger balances.

---

Do not use journal or subledger level reporting currencies if your organization has only an infrequent need to translate your financial statements to your parent company's currency for consolidation purposes. Standard translation functionality meets this need. Consider using journal or subledger level reporting currencies when any of the following conditions exist.

- You operate in a country whose unstable currency makes it unsuitable for managing your business. As a consequence, you need to manage your business in a more stable currency while retaining the ability to report in the unstable local currency.
- You operate in a country that is part of the European Economic and Monetary Union (EMU), and you choose to account and report in both the European Union currency and your National Currency Unit (NCU).

---

**Note**

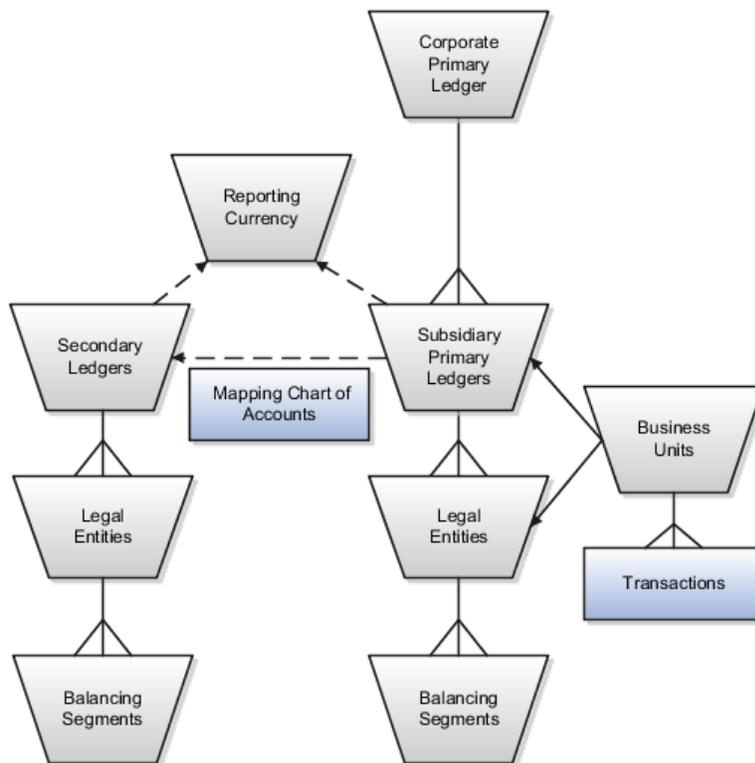
The second option is rare since most companies have moved beyond the initial conversion to the EMU currency. However, future decisions could add other countries to the EMU, and then, this option would again be used during the conversion stage.

---

## Financial Ledgers: How They Fit Together

Oracle Fusion Applications is an integrated suite of business applications that connects and automates the entire flow of the business process across both front and back office operations and addresses the needs of a global enterprise. The process of designing the enterprise structure, including the accounting configuration, is the starting point for an implementation. This process often includes determining financial, legal, and management reporting requirements, setting up primary and secondary ledgers, making currency choices, and examining consolidation considerations.

This figure shows the enterprise structure components and their relationships to each other. Primary ledgers are connected to reporting currencies and secondary ledgers to provide complete reporting options. Legal entities are assigned to ledgers, both primary and secondary, and balancing segments are assigned to legal entities. Business units must be connected to both a primary ledger and a default legal entity. Business units can record transactions across legal entities.



### Primary Ledgers

A primary ledger is the main record-keeping ledger. Create a primary ledger by combining a chart of accounts, accounting calendar, ledger currency,

and accounting method. To determine the number of primary ledgers, your enterprise structure analysis must begin with determining financial, legal, and management reporting requirements. For example, if your company has separate subsidiaries in several countries worldwide, create multiple primary ledgers representing each country with the local currency, chart of accounts, calendar, and accounting method to enable reporting to each country's legal authorities.

If your company just has sales in different countries, with all results being managed by the corporate headquarters, create one primary ledger with multiple balancing segment values to represent each legal entity. Use secondary ledgers or reporting currencies to meet your local reporting requirements, as needed. Limiting the number of primary ledgers simplifies reporting because consolidation is not required. Other consideration such as corporate year end, ownership considerations, and local government regulations, also affect the number of primary ledgers required.

## Secondary Ledgers

A secondary ledger is an optional ledger linked to a primary ledger. A secondary ledger can differ from its related primary ledger in chart of accounts, accounting calendar, currency, accounting method, or ledger processing options. Reporting requirements, for example, that require a different accounting representation to comply with international or country-specific regulations, create the need for a secondary ledger.

Below are scenarios and required action for different components in primary and secondary ledgers:

- If the primary and secondary ledgers use different charts of accounts, the chart of accounts mapping is required to instruct the system how to propagate journals from the source chart of accounts to the target chart of accounts.
- If the primary and secondary ledgers use different accounting calendars, the accounting date and the general ledger date mapping table will be used to determine the corresponding non-adjusting period in the secondary ledger. The date mapping table also provides the correlation between dates and non-adjusting periods for each accounting calendar.
- If the primary ledger and secondary ledger use different ledger currencies, currency conversion rules are required to instruct the system on how to convert the transactions, journals, or balances from the source representation to the secondary ledger.

Note: Journal conversion rules, based on the journal source and category, are required to provide instructions on how to propagate journals and types of journals from the source ledger to the secondary ledger.

## Reporting Currencies

Reporting currencies are the currency you use for financial, legal, and management reporting. If your reporting currency is not the same as your ledger currency, you can use the foreign currency translation process or reporting currencies functionality to convert your ledger account balances in your reporting currency. Currency conversion rules are required to instruct the system on how to convert the transactions, journals, or balances from the source representation to the reporting currency.

## Legal Entities

Legal entities are discrete business units characterized by the legal environment in which they operate. The legal environment dictates how the legal entity should perform its financial, legal, and management reporting. Legal entities generally have the right to own property and the obligation to comply with labor laws for their country. They also have the responsibility to account for themselves and present financial statements and reports to company regulators, taxation authorities, and other stakeholders according to rules specified in the relevant legislation and applicable accounting standards. During setup, legal entities are assigned to the accounting configuration, which includes all ledgers, primary and secondary.

## Balancing Segments

You assign primary balancing segment values to all legal entities before assigning values to the ledger. Then, assign specific primary balancing segment values to the primary and secondary ledgers to represent nonlegal entity related transactions such as adjustments. You can assign any primary balancing segment value that has not already been assigned to a legal entity. You are allowed to assign the same primary balancing segment values to more than one ledger. The assignment of primary balancing segment values to legal entities and ledgers is performed within the context of a single accounting setup. The Balancing Segment Value Assignments report is available to show all primary balancing segment values assigned to legal entities and ledgers across accounting setups to ensure the completeness and accuracy of their assignments. This report allows you to quickly identify these errors and view any unassigned values.

## Business Units

A business unit is a unit of an enterprise that performs one or many business functions that can be rolled up in a management hierarchy. When a business function produces financial transactions, a business unit must be assigned a primary ledger, and a default legal entity. Each business unit can post transactions to a single primary ledger, but it can process transactions for many legal entities. Normally, it will have a manager, strategic objectives, a level of autonomy, and responsibility for its profit and loss. You define business units as separate tasks generally done after the accounting setup steps.

The business unit model:

- Allows for flexible implementation
- Provides a consistent entity for controlling and reporting on transactions
- Enables sharing of sets of reference data across applications

For example, if your company requires business unit managers to be responsible for managing all aspects of their part of the business, then consider using two balancing segments, company and business unit to enable the production of business unit level balance sheets and income statements.

Transactions are exclusive to business units. In other words, you can use business unit as a securing mechanism for transactions. For example, if you have an export business that you run differently from your domestic business, use business units to secure members of the export business from seeing the transactions of the domestic business.

## Creating Primary Ledgers: Example

Create a primary ledger as your main record-keeping ledger. Like any other ledger, a primary ledger records transactional balances by using a chart of accounts with a calendar, currency, and accounting rules implemented in an accounting method. The primary ledger is closely associated with the subledger transactions and provides context and accounting for them.

### Scenario

Your company, InFusion Corporation is implementing Oracle Fusion Applications. You have been assigned the task of creating a primary ledger for your InFusion America entity.

1. Navigate to the **Define Accounting Configurations** task list and open **Manage Primary Ledgers** from within your implementation project. Click the **Go to Task**.
2. Click the **Create** icon.
3. Enter the following values:

Field	Value
Name	InFusion America
Description	InFusion America primary ledger for recording transactions.
Chart of Accounts	InFusion America Chart of Accounts
Accounting Calendar	Standard Monthly
Currency	USD
Accounting Method	Standard Accrual

4. Click **Save and Edit Task List** to navigate back to the accounting configuration task list.

---

### Note

You cannot change the chart of accounts, accounting calendar, or currency for your ledger after you save your ledger.

---

## Specifying Ledger Options: Worked Example

This example demonstrates specifying the ledger options for your primary ledger. Your company, InFusion Corporation, is a multinational conglomerate that operates in the United States (US) and the United Kingdom (UK). InFusion has purchased an Oracle Fusion enterprise resource planning (ERP) solution including Oracle Fusion General Ledger and all of the Oracle Fusion subledgers.

After completing your InFusion America Primary Ledger, select **Specify Ledger Options** under the Define Accounting Configuration task list on the Functional Setup Manager page.

---

## Note

Both primary and secondary ledgers are created in the same way and use the same user interface to enable their specific ledger options.

---

## Reviewing General Region Options

1. Accept the **Name** and **Description** defaults for the ledger selected.
2. Review the **Currency** and **Chart of Accounts** for the specified ledger, which are automatically populated.

## Setting Accounting Calendar Region Options

1. Review the **Accounting Calendar** that defaults from your ledger.
2. Select Jan-2011 as the **First Open Period** for your ledger.

Important: Select a period after the first defined period in the ledger calendar to enable running translation. You cannot run translation in the first defined period of a ledger calendar. In this example, your calendar began with Jan-2010.

3. Enter 3 for the **Number of Future Enterable Periods**.

Any value between 0 and 999 periods can be specified to permit entering journals but not posting them in future periods. Minimize the number of open and future periods to prevent entry in the wrong period.

## Selecting the Subledger Accounting Region Options

1. Accept the default **Accounting Method** from your ledger.
2. Select US American English as your **Journal Language**.

## Completing the Period Close Region Options

1. Enter your **Retained Earnings Account**:  
101-00-31330000-0000-000-0000-0000.

This account is required for the General Ledger to perform the movement of revenue and expense account balances to this account at the end of the accounting year.

2. Enter your **Cumulative Translation Adjustment Account**:  
101-00-31350000-0000-000-0000-0000.

Note: The Cumulative Translation Adjustment (CTA) account is required for ledgers running translation.

3. Do not enter a **Default Period End Rate Type** or **Default Period Average Rate Type**.

The values entered here are used as the default for balance level reporting currency processing. InFusion America Primary Ledger is using the subledger level reporting currency processing.

## Specifying the Journal Processing Region Options

1. Specify the Balance options as outlined in the following table.

Option	Setting
Enable Suspense	General Ledger
Default Expense Account	101-00-98199999-0000-000-0000-0000
Rounding Account	101-10-98189999-0000-000-0000-0000
Entered Currency Balancing Account	101-10-98179999-0000-000-0000-0000
Balancing Threshold Percent	10

2. Click all the following Entry options listed in the table.

Option	Description
Enable journal approval	Click to enable journal approval functionality. Approval rules must be created in the Oracle Fusion Approvals Management (AMX).
Notify when prior period journal	Notify the user when a prior period date is selected on a journal entry.
Allow mixed and statistical journals	Enter both monetary and statistical amounts on the same line in a journal entry.
Validate reference date	Requires a reference date in an open or future enterable period.

3. Click the **Separate journals by accounting date during journal import** for the Import option to create individual journal entries for each accounting date.
4. For the Reversal options, select InFusion America Accrual Set from the list of values in the **Journal Reversal Criteria Set** field and click the **Launch AutoReverse after open period** to reverse accrual journal entries automatically when a new period is opened.
5. Click the **Enable intercompany accounting** for the Intercompany option to enable automatic balancing by the application for primary, second, and third balancing segments (if implemented) on intercompany journal entries and transactions.

Note: To complete the intercompany accounting functionality, you must define intercompany rules.

## FAQs for Specify Ledger Options

### What happens if I change the cumulative adjustment account?

To avoid data corruption, your cumulative adjustment account (CTA) can only be changed if you first perform the following set of steps:

- Purge all translated balances
- Change the CTA account
- Rerun translation

### **What happens if I change the retained earnings account?**

To avoid data corruption, your retained earnings account can only be changed if you first perform the following set of steps:

- Enter and post journals to bring the ending balances for your income statement accounts to zero at the end of each accounting year
- Purge actual translated balances
- Update the retained earnings account
- Reverse the journal entries use to bring the ending account balances to zero and rerun translation

## **Assigning Legal Entities and Balancing Segments: Examples**

Optionally, assign legal entities and balancing segments to your accounting configuration.

### **Assign Legal Entities**

Assign one or more legal entities to your configuration by following these steps:

1. Navigate to the **Assign Legal Entities** task. Click the **Go to Task**.
2. Click the **Select and Add** icon.
3. Click **Search**. Select your legal entities.
4. Click **Apply**. Click **Done**.
5. Click **Save and Close**.

### **Assign Balancing Segments to Legal Entities**

Assign balancing segment values to your legal entities by following these steps:

1. Navigate to the **Assign Balancing Segment Values to Legal Entities** task. Click the **Go to Task**.
2. Click the **Create** icon.
3. Select the balancing segment value. Optionally, add a Start Date.
4. Click **Save and Close** to close the create page.
5. Click **Save and Close**.

### **Assign Balancing Segments to Ledgers**

Assign balancing segment values directly to your ledger by following these steps:

1. Navigate to the **Assign Balancing Segment Value to Ledger** task. Click the **Go to Task**.
2. Select the balancing segment value.
3. Optionally enter a start date.
4. Click **Save and Close**.

---

**Note**

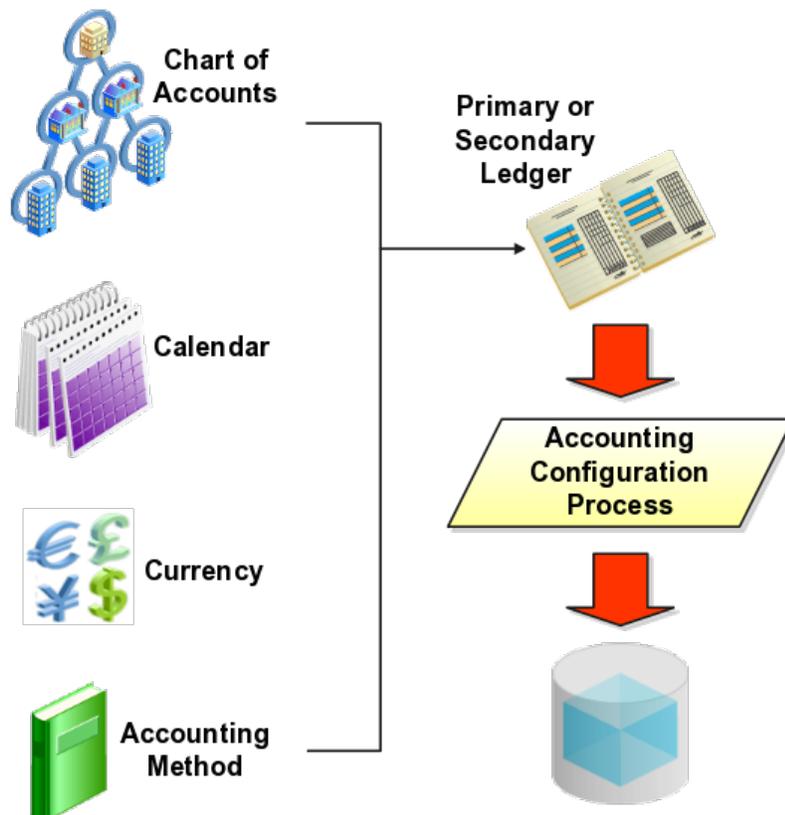
The balancing segment values that are assigned to the ledger represent nonlegal entity transactions, such as adjustments. If you use legal entities, you must assign balancing segment values to all legal entities before assigning values to the ledger. The only available balancing segment values that can be assigned to ledgers are those not assigned to legal entities.

---

## Review and Submit Accounting Configuration

### Oracle Fusion General Ledger Balances Cube: Overview

A balances cube is an online analytical application (OLAP) database that maintains financial balances in a multidimensional database. In Oracle Fusion General Ledger a new balances cube is created when an accounting configuration is submitted for a primary or secondary ledger that uses to a new unique combination of chart of accounts and calendar.



A new balances cube is also created when a secondary ledger is added to an existing accounting configuration and uses a new unique combination of chart of accounts and calendar. The balances cubes are named after the chart of accounts they contain.

A balances cube:

- Stores your financial balances in a multidimensional enable real time, interactive financial reporting and analysis.
- Preaggregates your balances at every possible point of summarization, thus ensuring immediate access to financial data and eliminating the need for an external data warehouse for financial reporting.
- Is uniquely identified by the combination of a chart of accounts and an accounting calendar. Average balances are tracked in a separate balances cube.
- Is automatically synchronized by the following general ledger processes: posting, open period, and translation.
- Consists of a set of defining business entities called dimensions. Dimensions in a cube determine how data is accumulated for reporting and analytical purposes
- Are referred to as an application or database connection in the user interfaces for:
  - Financial Reports
  - Smart View
  - Calculation Manager

---

**Note**

Account balances were maintained in relational tables in Oracle E-Business Suite General Ledger, The Oracle Fusion General Ledger updates balances in real time to the relational tables and the General Ledger Balances cubes. Most inquiry and reporting are now performed from the General Ledger Balances cubes and not from the relational tables.

---

## Standard Balances Cube Dimensions: Explained

A balances cube consists of a set of defining business entities called dimensions. This table details the dimensions that are available for creating financial reports, Smart View queries, and allocations using multidimensional cubes.

Dimension	Description	Example
Accounting Period	Based upon the calendar of the ledger or ledger set. Report on years, quarters, or periods.	<ul style="list-style-type: none"> <li>• 2012</li> <li>• Qtr-1</li> <li>• Jan-12</li> </ul>

Ledger or Ledger Set	Used to select a ledger for the reporting. Multiple ledgers may be in the same cube if they share a common chart of accounts.	<ul style="list-style-type: none"> <li>• InFusion North America Ledger Set</li> <li>• InFusion US Primary Ledger</li> </ul>
Chart of Accounts Segments	Uses a separate dimension for each of the segments from the charts of accounts. Organized by hierarchy. A default hierarchy is provided that includes all detail segment values. Hierarchies published in the Publish Account Hierarchies user interface are included.	<ul style="list-style-type: none"> <li>• Company: InFusion America: 101</li> <li>• Cost Center: Sales: 400</li> <li>• Account: Cash: 1110</li> </ul>
Scenario	<p>Indicates if the balances represented are actual or budget amounts. Allocation related dimensions are seeded members and required for allocation solutions. Allocation dimension are not used directly by end users.</p> <p>Budget scenario dimension members are user defined in the Oracle Fusion Applications value set called Accounting Scenario and appear in the cube after running the <b>Create Scenario Dimension Members</b> process.</p>	<ul style="list-style-type: none"> <li>• Budget 2012</li> <li>• Actuals</li> <li>• Forecast 2013</li> </ul>
Balance Amount	Indicates if the value is the beginning balance, period activity, or ending balance. Debit, Credit, and Net amounts are available for reporting.	<ul style="list-style-type: none"> <li>• Beginning Balance (Dr, Cr, or Net)</li> <li>• Period Activity (Dr, Cr, or Net)</li> <li>• Ending Balance (Dr, Cr, or Net)</li> </ul>
Amount Type	Indicates whether the amounts represent Base, Period to Date, Quarter to Date, or Year to Date.	<ul style="list-style-type: none"> <li>• Base</li> <li>• PTD: Period to Date</li> <li>• QTD: Quarter to Date</li> <li>• YTD: Year to Date</li> </ul>
Currency	Used to select the desired currency for the balances.	<ul style="list-style-type: none"> <li>• All ISO Currencies</li> <li>• USD: US Dollar</li> <li>• JPY: Japanese Yen</li> </ul>
Currency Type	Used to select the currency type of the balances.	<ul style="list-style-type: none"> <li>• Total</li> <li>• Entered</li> <li>• Converted From (for each ISO currency)</li> </ul>

---

**Note**

Dimensions are seeded and new ones cannot be added.

---

## Creating GL Balances Cubes: Examples

There are two types of Oracle Fusion General Ledger Balances cubes: Standard Balances cubes and Average Balances cubes.

### Standard Balances Cubes

A new standard balances cube is created whenever an accounting configuration is submitted for a ledger, either primary or secondary, that uses a new unique combination of chart of accounts and accounting calendar. Cubes are named after the chart of accounts.

For example, the chart of accounts, InFusion US Chart of Accounts has a related cube entitled, InFusion US Chart of Accounts. If a chart of accounts is used by multiple ledgers with different calendars, the cube names are distinguished by appending a number to their names.

For example, if InFusion US Chart of Accounts is used by two different ledgers, each of which has a different accounting calendar, one with a standard calendar year ending December 31st and the other with a fiscal year ending May 31st, two cubes are created. The cubes are named InFusion US Chart of Accounts and InFusion US Chart of Accounts 2.

### Average Balances Cubes

Average balances cubes use different dimensions than the standard balances cubes therefore require their results be stored in separate cubes. If the average balances option is enabled for a ledger, a second average balances cube is automatically created based upon the same criteria of a unique combination of chart of accounts and accounting calendar. Average balances cubes are named with ADB (average daily balances) plus the name of the associated chart of accounts.

For example, for a chart of accounts, InFusion US Chart of Accounts, the average balances cube name is **ADB InFusion US Chart of Accounts**. Numbers are appended to the name when more than one calendar is used by the same chart of accounts. The numbering is determined and shared with the related standard balances cubes. The standard cube called **InFusion US Chart of Accounts 3** has a corresponding average balance cube entitled **ADB InFusion US Chart of Accounts 3**.

## Customer Specific GL Balances Cube Dimension Values: Points to Consider

In creating your cube, take in to consideration the dimensions that are you, the customer, define and those that are partially or completely predefined by Oracle Fusion Applications.

- Two dimensions, Chart of Accounts and Scenario, have customer specific dimension values and require user procedures to populate the cube

- Accounting Period, Currency, and Ledger dimensions are customer specific, but are automatically created in the cubes from the changes made in the applicable user interfaces.
- Other dimensions, such as Amount Type and Balance Amount, have only predefined members.

The following are points to consider in setting up and populating cubes for the Chart of Accounts and Scenario dimension members.

### Create Account Hierarchies

- Account hierarchies for your chart of account dimensions are created in the **Manage Account Hierarchies** page or the **Manage Trees and Trees Versions** page after setting up your value sets, chart of accounts, and values.
- Create account hierarchies (trees) to identify managerial, legal, or geographical relationships between your value set values.
- Define date-effective tree versions to reflect organizational changes within each hierarchy over time.
- The tree version must be in an active or inactive status to be published. Draft versions cannot be published.

---

#### Note

From your implementation project, **Navigate > Define Common Applications Configuration > Define Enterprise Structures > Define Financial Reporting Structures > Define Chart of Accounts > Manage Account Hierarchies > Go to Task.**

---

### Publish Account Hierarchies

Next, after defining the tree versions, publish account hierarchies (tree versions) to the cube. Before publishing, set the following profile option: **Display Only Segment Value Description in the Cube Alias**. Consider the following points when setting the profile option.

- The member alias displayed in the Oracle Fusion General Ledger balances cubes.
  - Yes displays only the segment value description, for example: Cash.
  - No displays value plus the segment value description, for example: 1110 Cash.
- Default value is No.
- No is necessary when there are duplicate segment value descriptions across all dimensions. The descriptions are stored in the Alias in the cube. Aliases must be unique across all dimensions in the cube. For example, if the description West existed in the Cost Center value set and the Location value set, the entire cube creation process fails.

- If duplicate segment value descriptions exist, the cube can be created with the profile set to No. At anytime the value of the profile can be changed to Yes after the duplicates are removed. At that time, the account hierarchies (tree versions) must be published or republished to the cube.
- If you set the profile value to Yes, and rebuild a cube from a process run in Enterprise Scheduler or create a cube by submitting an accounting configuration and duplicate descriptions exist, the create cubes process errors. You have two options at that point:
  - Set profile value to No and run the Create Cubes process or if you have a new ledger with a new chart of accounts and accounting calendar combination, submit the accounting configuration.
  - Remove the duplicates across all dimensions, not just a single segment. Then run the **Create Cubes** or resubmit the accounting configuration.

From your implementation project, **Navigate > Define Common Applications Configuration > Define Enterprise Structures > Define Financial Reporting Structures > Define Chart of Accounts > Publish Account Hierarchies > Go to Task**. Use the **Publish Account Hierarchies** page to search and publish account hierarchies.

- Select the **Publish** check box. This is indicator of what to include in balances cube by selecting the check box or what you don't want to include in balances cube by removing the check from the check box.
- Select the rows. Functionality allows for select multiple rows to be selected.
- Select the **Publish** button to update the balances cubes. A process is generated.
- **Navigator > Tools > Scheduled Processes** to monitor the process.

---

#### Note

Use **Smart View** to verify that the account hierarchies were correctly published.

---

### Define Scenario Dimension Members

The customer specific Scenario dimension members are defined in the **Manage Value Sets** and **Manage Values** pages in the value set called **Accounting Scenario**. Any customer specific Scenario dimension is included in all balances cubes.

Best practice is to setup the customer specific Scenario members in this value set before you create your first ledger and run the **Accounting Configuration** process. The **Accounting Configuration** process generates your balances cubes.

If the cubes already existed, you can run the **Create Scenario Dimension Members** process to update the balances cubes for the new members.

---

#### Note

If you are on a release before the **Create Scenario Dimension Members** process is available, you have to rebuild the cubes with the **Create Cubes** process to add the Scenario dimension in the cube. Follow the guidelines for creating cubes before running process.

## Default Dimension Values in Reporting: Explained

For Standard Balances Cube dimensions, there are default values that if used in Smart View and on financial reports created in Financial Reporting Studio cause #MISSING to appear when balances are returned on a report output. If #MISSING appears, check that all dimensions are properly set. Particularly, check the Accounting Period, Ledger, Scenario, and Currency dimensions, which all must have a value selected other than the default top level value called Gen1.

The following table lists the available and default dimension values as well as providing guidance on selecting the correct dimensions.

Dimension	Possible Values	Default Value	Additional Information
Accounting Period	Years, quarters, and period	Accounting Period = Accounting Period (Gen1)	You must always select an accounting period for each financial report including User Point of View, Smart View query, or allocation including Point of View. If you do not specify a valid Accounting Period, the financial reports, Smart View queries, and Account Inspector displays #MISSING.
Ledger	Includes ledgers and ledger sets	Ledger = Ledger (Gen1)	If you do not specify a specific Ledger or Ledger Set, the financial reports, Smart View queries, and Account Inspector queries display #MISSING
Chart of accounts dimensions		Highest level (Gen1)	There is a separate dimension for each segment of the chart of accounts used by the cube. Each segment has a default account hierarchy that includes all the detail values for the segment but not parent values. Only account hierarchies (tree versions) published from the <b>Publish Account Hierarchies</b> user interface are available in the cube.

Scenario		Scenario = Scenario represents the sum of all values: Actual + Allocated + Total for Allocations + Budget + Forecast. Select a value for this dimension.  <b>Note</b> Must always select a Scenario dimension.	Seeded values are Actual, Allocated, and Total for Allocated. Additional scenario values for Budget, Forecast Q1, and Forecast Q2 may be available if they have been added to the Accounting Scenario value set. These additional values will be published to every cube.
Balance Amount	<ul style="list-style-type: none"> <li>Beginning Balance DR, CR, or Net</li> <li>Period Activity DR, CR, or Net</li> <li>Ending Balance DR, CR, or Net</li> <li>Balance Amount (same as Ending Balance)</li> </ul>	Balance Amount = Balance Amount (Gen1) which is the equivalent of Ending Balance.	
Amount Type	Base, PTD, QTD, YTD	Amount Type = Amount Type (Gen1) which is the equivalent of Base.	Base is necessary because this is the value used to store from posting all balances at the lowest level. PTD, QTD, and YTD are calculated values.
Currency	All ISO currencies (250+).	Highest level (Gen1)	Similar to Accounting Period and Ledger, there may not be an appropriate default for Currency since different Ledgers have different ledger currencies. If you don't specify a valid currency in a financial report, Smart View query, or allocation, you will get a result = #MISSING.
Currency Type	Total, Entered, and Converted from for each ISO currency = each ISO currency.	Currency Type = Currency Type (Gen1), which is the equivalent of Total.	

### Average Balances Cubes Dimensions

The following dimensions are included in the average balances cube in this order. Most comments from standard cube are applicable below unless stated.

- Accounting Period: Valid values are years, quarters, accounting period and day, which is equivalent to accounting date.
- Ledger

- Separate dimension for each Chart of accounts segment
- Scenario
- Amount Type valid values are: PATD, QATD, and YATD
- Currency
- Currency Type

## GL Balances Cube Terminology: Explained

Levels and generations are cube terminology used to describe hierarchies in Oracle Fusion General Ledger balances cubes.

### Levels

Levels are used to describe hierarchy levels. Levels are numbered from the lowest hierarchy level. For example, the detail chart of accounts segment values are Level 0. The immediate parent is Level 1; the next parent is Level 2.

### Generations

Generations (Gen) describe hierarchy levels starting with the top of the hierarchy and moving down through the generations of the same.

An example for generations is as follows:

- Accounting Period = Accounting Period is Gen 0
  - Year is Gen 1
    - Qtr is Gen 2
      - Period is Gen 3
- Ledger = Ledger is Gen 1
  - All Ledgers is Gen 2
    - VF USA is Gen 3
  - Ledger Set (any ledger set) is Gen 2
- Company = Company is Gen 1
  - All Company Values, for example detail values, is Gen 2
    - 11010 is Gen 3
  - Tree 1 V1 is Gen 2
    - [Tree 1 V1].[10000] is Gen 3
      - [Tree 1 V1].[10000].[11000] is Gen 4
  - Tree 2 V1 is Gen 2

- [Tree 2 V1].[10000] is Gen 3
- [Tree 1 V1].[10000].[11000] is Gen 4

---

**Note**

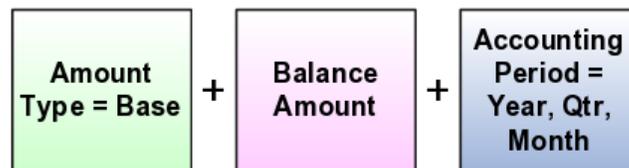
Similar levels and generations apply to the other dimension, including chart of accounts dimensions and those that are not chart of accounts dimensions.

---

## Using Dimension Values in Reporting: Examples

By using various combinations of selections for the Accounting Period, Balance Amount, and Amount Type dimensions, you can derive different amounts to meet financial reporting requirements. In some cases, more than one combination of query values can return the desired information.

There is some duplication in the calculations for the balances cube to ensure complete reporting requirements.



Many reporting needs can be completed using the Balance Amount dimension, Amount Type equal to Base, and specifying the Accounting Period equal to Year, Quarter, or Month.

However, the Amount Type dimension is still required for the following reasons:

- To get the complete QTD (Quarter to Date) reporting requirements for the first and second month in a quarter can only be achieved using QTD and Accounting Period = Month.
- PTD (Period to Date) and YTD (Year to Date) are also available to ensure more consistency reporting with the E-Business Suite Release 12.

### Obtaining Quarter Information

This example shows how to obtain quarter information.

- QTD for the first and second months of a quarter can only be achieved using Amount Type dimension equal to QTD.
- The end of quarter information can be derived from the Accounting Period dimension equal to Quarter or with the Amount Type equal QTD.

### Obtaining End of Year Information

This example shows how to obtain end of year information.

- Accounting Period equal to Year and Balance Amount equal to Ending Balance and Amount Type equal to Base or YTD.
- Accounting Period equal to a specific month and Balance Amount equal to Period Activity and Amount Type equal to YTD.

- When the Balance Amount equals the Balance Amount or the Balance Amount equals the Ending Balance, this results in an Ending Balance.

---

**Note**

Ending Balance is always the ending balance regardless of Amount Type member setting or Accounting Period member setting (period, quarter or year).

---

## Standard Balances: Example

You have loaded your Oracle E-Business Suite Release 12 balances into your new Oracle Fusion Standard Balances Cube. Now you want to query the results to verify that the data was loaded correctly.

### Scenario

The following two tables show the amounts loaded into the balances cube from your R12 balances and the calculated balances for ending balance, year to date (YTD), and quarter to date (QTD). The first table is the balance sheet based balances and the second table is for income statement based balances.

### Oracle E-Business Suite R12 Balances

#### Balance Sheet Balances

Month	Beginning Balance Loaded to Cube	Activity Loaded to Cube	Ending Balance	Calculate YTD	QTD
Jan	200	20	220	20	20
Feb	220	30	250	50	50
Mar	250	25	275	75	75
Apr	275	50	325	125	50
May	325	40	365	<b>165</b>	90
Jun	365	45	410	210	135
Jul	410	100	510	310	100
Aug	510	200	710	510	300
Sep	710	140	850	650	440
Oct	850	150	1000	800	150
Nov	1000	100	1100	900	250
Dec	1100	400	1500	1300	650

#### Income Statement Balances

Month	Beginning Balance Loaded to Cube	Activity Loaded to Cube	Ending Balance	Calculate YTD	QTD
Jan	0	70	70	70	70
Feb	70	20	90	90	90
Mar	90	30	120	120	120

Apr	120	100	220	220	100
May	220	200	420	420	300
Jun	420	250	670	670	550
Jul	670	50	720	720	50
Aug	720	300	1020	1020	350
Sep	1020	130	1150	1150	480
Oct	1150	110	1260	1260	110
Nov	1260	200	1460	1460	310
Dec	1460	500	1960	1960	810

### Analysis

The following two tables show the results available from Oracle Fusion General Ledger from your R12 loaded and aggregated balances in the balances cube. The first table is the balance sheet based balances and the second table is for income statement based balances.

#### Balance Sheet Balances

Accounting Period	Balance Amount	Amount Type	Value	Comments	R12 Amount Type Equivalent
May	Beginning Balance	YTD	200	Jan Beginning Balance	
May	Period Activity	YTD	165	Jan to May Activity	YTD-Actual as of May
May	Ending Balance	YTD	365	May Ending Balance	YTD-Actual as of May
May	Beginning Balance	QTD	275	Apr Beginning Balance	
May	Period Activity	QTD	90	Apr-May Activity	QTD-Actual as of May
May	Ending Balance	QTD	365	May Ending Balance	QTD-Actual as of May
May	Beginning Balance	PTD	325	May Beginning Balance	
May	Period Activity	PTD	40	May Activity	PTD-Actual as of May
May	Ending Balance	PTD	365	May Ending Balance	PTD-Actual as of May
May	Beginning Balance	Base	325	May Beginning Balance	
May	Period Activity	Base	40	May Activity	PTD-Actual as of May
May	Ending Balance	Base	365	May Ending Balance	YTD-Actual as of May
Q2	Beginning Balance	QTD	275	April Beginning Balance	

Q2	Period Activity	QTD	135	Always Apr-Jun	QTD-Actual as of Jun
Q2	Ending Balance	QTD	410	Ending Balance always as of Jun	QTD-Actual as of Jun
Q2	Ending Balance	YTD	410	Ending Balance always as of Jun	YTD-Actual as of Jun
Q2	Beginning Balance	Base	275	Apr Beginning Balance	
Q2	Period Activity	Base	135	Always Apr-Jun	QTD-Actual as of Jun
Q2	Ending Balance	Base	410	Ending Balance always as of Jun	YTD-Actual as of Jun
Year	Beginning Balance	YTD	200	Jan Beginning Balance	
Year	Period Activity	YTD	1300	Jan-Dec Activity	YTD-Actual as of Dec
Year	Ending Balance	YTD	1500	Ending Balance always as of Dec	YTD-Actual as of Dec
Year	Beginning Balance	Base	200	Jan Beginning Balance	
Year	Period Activity	Base	1300	Jan-Dec Activity	YTD-Actual as of Dec
Year	Ending Balance	Base	1500	Ending Balance always as of Dec	YTD-Actual as of Dec

#### Income Statement Balances

Accounting Period	Balance Amount	Amount Type	Value	Comments	R12 Amount Type Equivalent
May	Beginning Balance	YTD	0	Jan Beginning Balance	
May	Period Activity	YTD	420	Jan to May Activity	YTD-Actual as of May
May	Ending Balance	YTD	420	May Ending Balance	YTD-Actual as of May
May	Beginning Balance	QTD	120	Apr Beginning Balance	
May	Period Activity	QTD	300	Apr-May Activity	QTD-Actual as of May
May	Ending Balance	QTD	420	May Ending Balance	QTD-Actual as of May
May	Beginning Balance	PTD	220	May Beginning Balance	
May	Period Activity	PTD	200	May Activity	PTD-Actual as of May
May	Ending Balance	PTD	420	May Ending Balance	PTD-Actual as of May

May	Beginning Balance	Base	220	May Beginning Balance	
May	Period Activity	Base	200	May Activity	PTD-Actual as of May
May	Ending Balance	Base	420	May Ending Balance	YTD-Actual as of May
Q2	Beginning Balance	QTD	120	Apr Beginning Balance	
Q2	Period Activity	QTD	550	Period Activity Always Apr-Jun	QTD-Actual as of Jun
Q2	Ending Balance	QTD	670	Ending Balance always as of Jun	QTD-Actual as of Jun
Q2	Ending Balance	YTD	670	Ending Balance always as of Jun	YTD-Actual as of Jun
Q2	Beginning Balance	Base	120	Apr Beginning Balance	
Q2	Period Activity	Base	550	Period Activity always Apr-Jun	QTD-Actual as of Jun
Q2	Ending Balance	Base	670	Ending Balance always as of Jun	YTD-Actual as of Jun
Year	Beginning Balance	YTD	0	Jan Beginning Balance (always zero for Income Statement)	
Year	Period Activity	YTD	1960	Jan-Dec Activity	YTD-Actual as of Dec
Year	Ending Balance	YTD	1960	Ending Balance always as of Dec	YTD-Actual as of Dec
Year	Beginning Balance	Base	0	Jan Beginning Balance (always zero for Income Statement)	
Year	Period Activity	Base	1960	Jan-Dec Activity	YTD-Actual as of Dec
Year	Ending Balance	Base	1960	Ending Balance always as of Dec	YTD-Actual as of Dec

## Invalid Balance Cube Dimension Combinations: Explained

By using various combinations of selections for the Accounting Period, Balance Amount, and Amount Type dimensions, you can derive different amounts to meet financial reporting requirements combinations.

Balances cubes do not return data for these combinations:

Accounting Period	Balance Amount	Amount Type
Year	Beginning Balance	QTD

Year	Period Activity	QTD
Year	Ending Balance	QTD
Year	Beginning Balance	PTD
Year	Period Activity	PTD
Year	Ending Balance	PTD
Q2	Beginning Balance	YTD
Q2	Period Activity	YTD
Q2	Beginning Balance	PTD
Q2	Period Activity	PTD
Q2	Ending Balance	PTD

## GL Balance Cube Related Processes Run from User Interfaces

This list describes the processes run from the application pages (user interfaces) used to create, rebuild, publish, and maintain Oracle Fusion General Ledger balances cubes, including automatically run child processes.

### Create Cube from Accounting Configuration Page

Navigate > Setup and Maintenance work area > Define Common Applications Configuration > Define Ledgers > Define Accounting Configuration > Review and Submit Accounting Configuration

Parent Process	Child Process	Description	Cube Type
Create Accounting Configuration		Updates the underlying table and permanently saves the settings after the accounting configuration has been completed. Some settings cannot be changed or deleted.	Standard
	Create Accounting Configuration: Create Cubes	Creates the accounting configuration and a balances cube when setup is completed.	Standard and ADB, if enable, in same request

### Publish Chart of Accounts from Publish Account Hierarchies Page

Navigate > Setup and Maintenance work area > Define Common Applications Configuration > Define Enterprise Structures > Define Financial Reporting Structures > Define Chart of Counts > Publish Account Hierarchies

Parent Process	Child Process	Description	Cube Type
Publish Chart of Account Hierarchies		Creates the structural hierarchy for the chart of accounts.	Standard and ADB, if enable, in same request
	Publish Chart of Account Hierarchy Versions	One runs for each combination of value set and chart of accounts.	Standard and ADB, if enable, in same request

### Create Ledger Set Processes from Ledger Set Page

Navigate > Setup and Maintenance work area > Define Common Applications Configuration > Define General Ledger Options > Manage Ledger Sets

Parent Process	Child Process	Description	Cube Type
Create Ledger Set		Updates the underlying tables and creates the ledger set.	Standard and ADB, if enable, in same request
	Create Accounting Configuration: Process Accounting Configuration	Runs processes necessary to process accounting configuration and ledger set data.	Not applicable to the Cube
	Create Ledger Set Dimension Members	Creates and updates ledger set members in a balances cube.	Standard and ADB, if enable, in same request

### Other Requests Submitted from Pages or Processes

Parent Process	Child Process	Description	Cube Type	Notes
Create Ledger Dimension Member for Reporting Currency		Creates ledger dimension member for reporting currency.	Standard and ADB, if enable, in same request	Navigate > Setup and Maintenance work area > Define Common Applications Configuration > Define Ledgers > Define Accounting Configuration > Manage Reporting Currencies > Create Reporting Currency page, run for new reporting currencies added after accounting configuration has completed.
Update Ledger Dimension Member for Reporting Currency		Updates ledger dimension member for reporting currency.	Standard and ADB, if enable, in same request	Navigate > Setup and Maintenance work area > Define Common Applications Configuration > Define Ledgers > Define Accounting Configuration > Manage Reporting Currencies->Update Reporting Currency page, this job runs for new reporting currencies added after accounting configuration has completed.

	Publish Chart of Accounts Dimension Members: Detailed Values Only	Publishes chart of accounts dimension member changes to balances cubes and updates dimension members for any new or changed segment values.	Standard and ADB, if enable, in same request	Run only from the journal posting program when detailed values do not exist in the GL balances cube during posting.
--	---	---	--	---

## GL Balance Cube Related Processes Run in ESS

This list describes the processes used to create, rebuild, publish, and maintain Oracle Fusion General Ledger balances cubes, including automatically run child processes.

### Create Cube Processes Run in Oracle Enterprise Scheduler Service (ESS)

Parent Process	Child Process	Description	Cube Type
Create Cube		Creates or rebuilds the balances cube based on the combination of chart of accounts and accounting calendar.  Automatically runs the child processes below for standard and average daily balance (ADB) cubes, if enabled.	Standard
	Create Cube: Initialize Cube	Starts the process to import data into the balances cube.	Standard
	Create Cube: Create Calendar Dimension Members and Hierarchies	Creates the calendar dimension members and hierarchies for a balances cube.	Standard
	Create Cube: Create Ledger Dimension Members	Creates the ledger dimension members for a balances cube.	Standard
	Create Cube: Publish Chart of Accounts Dimension Members and Hierarchies	Publishes chart of accounts and hierarchy changes to balances cubes and updates dimension members for any new or changed segment values.	Standard
	Create Average Daily Balance Cube	Determines the amount needed to manually adjust general ledger account average balances to reflect the differences between the original and revalued customer open items.	Average Daily Balance (ADB)
	Create Cube: Initialize Average Balances Cube	Begins the process to import average balances into the balances cube.	ADB

	Create Cube: Create Daily Dimension Members and Hierarchies	Creates the daily calendar dimension members and hierarchies for a balances cube.	ADB
	Create Cube: Create Ledger Dimension Members	Creates the ledger dimension members for a balances cube.	ADB
	Create Cube: Publish Chart of Accounts Dimension Members and Hierarchies	Publishes chart of accounts and hierarchy changes to balances cubes and updates dimension members for any new or changed segment values.	ADB
	Create Cube: Transfer General Ledger Balances to Essbase	Transfer balances to balances cubes.	Standard and ADB, if enable, in same request

### Publish Cube Processes Run in ESS

Parent Process	Child Process	Description	Cube Type
Publish Chart of Accounts Dimension Members and Hierarchies		Publishes chart of accounts dimension member and hierarchy changes to balances cubes and updates dimension members for any new or changed segment values.	Standard and ADB, if enable, in same request
	Update Chart of Accounts Dimension Members and Hierarchies	Updates chart of accounts dimension member and hierarchy changes to Essbase.	Standard and ADB, if enable, in same request

### Transfer Cube Process Run in ESS

Parent Process	Description	Cube Type
Transfer General Ledger Balances to Essbase	Refreshes standard cube (and average balances cube if enabled) in the General Ledger balances cube.	Standard and ADB, if enable, in same request

### Other Cube Processes Run in ESS

Parent Process	Descriptions	Cube Type
Create Accounting Period Dimension for Standard Cube	Creates the accounting period dimension members.	Standard
Create Ledger Dimension Members	Creates and updates ledger dimension members including primary ledgers, secondary ledgers, reporting currencies, and ledger sets in the balances cubes.	Standard and ADB, if enable, in same request

Create Currency Dimension Members	Creates and updates all currencies in every balances cube.	Standard and ADB, if enable, in same request
Create Scenario Dimension Members	Creates and updates all scenario dimension members when new scenarios are created or existing scenarios are changed.	Standard and ADB, if enable, in same request
Create Accounting Period Dimension for Average Daily Balances Cube	Creates the accounting period dimension members in the average daily balances cube.	ADB
Create Rules XML File for BI Extender Automation		Standard and ADB, if enable, in same request

To run cube process, use the following steps the Scheduled Processes work area from the Navigator menu.

1. Click the **Schedule New Process** button
2. Search on the Process Name.
3. Enter the desired parameters.
4. Enter the desired process options and schedule.
5. Click Submit.

## FAQs for Review and Submit Accounting Configuration

### How can I secure GL balances cubes?

Secure GL balances cubes with chart of accounts dimension values, which use data access set and segment value security. Security restricts the display of data, not the selection of dimension values.

### How can I inquire on translated balances?

To inquire on translated balances, for example balance level reporting currency balances, query on Currency Type equal to Total and Currency equal to Translated Currency.

### When do the GL balances cubes need to be rebuilt?

In rare cases, you may need to rebuild the cubes. Carefully consider requirements and work with Oracle Support before rebuilding a cube. Use the on demand programs to rebuild dimension members for each dimension and to refresh balances to the cubes rather than rebuilding the cube. When you rebuild a cube, the process rebuilds both the standard and average balances cubes

To rebuild cubes, run the **Create Cubes** process. Enter values for the following parameters: **Chart of Accounts**, **Accounting Calendar**, and **Starting Period** for balances.

### How can I refresh balances in the GL balances cubes?

Run the **General Ledger Transfer Balances to Essbase** process. Select your Ledger or Ledger Set and Starting Period to refresh balances in the GL balance cubes.

# Manage Reporting Currencies

## Reporting Currency Balances: How They Are Calculated

Reporting currency balances, set at the journal or subledger level, are updated when journal entries that originate in Oracle Fusion General Ledger are posted and converted to your reporting currencies. This process includes General Ledger manual journals, periodic journals, and allocations, and at the subledger level, journals from Oracle Fusion Subledger Accounting and imported from sources other than your Oracle Fusion subledgers. When you post a journal in a ledger that has one or more reporting currencies defined, the posting process creates new journals converted to each of your reporting currencies and includes them in the same batch as the original journal with a status of Posted.

### Settings That Affect

Reporting currencies share a majority of the ledger options with their source ledger. For example, the reporting currency uses the same suspense account and retained earnings accounts as its source ledger. However, there are certain options that need to be set specifically for the reporting currencies. For example, reporting currencies are maintained at one of these three currency conversion levels:

- **Balance Level:** Only balances are maintained in the reporting currency using the General Ledger Translation process.
- **Journal Level:** Journal entries and balances are converted to the reporting currency by the General Ledger Posting process.
- **Subledger Level:** Subledger Accounting creates reporting currency journals for subledger transactions. General Ledger converts journals that originated in General Ledger or that are imported from sources other than the Oracle Fusion subledgers. The full accounting representation of your primary ledger is maintained in the subledger level reporting currency.

---

#### Note

Secondary Ledgers cannot use subledger level reporting currencies.

---

There are multiple dependencies between a reporting currency and its source ledger. Therefore, it is important that you complete your period opening tasks, daily journal or subledger level reporting currencies accounting tasks, and period closing tasks in the correct order. Some guidelines are presented in the table below.

Type	Task
Period Opening Tasks	Open the accounting period in both your ledger and reporting currencies before you create or import journals for the period. Converted journals are only generated in your reporting currency if the period is open or future enterable.

Daily Tasks	Enter the daily conversion rates to convert your journals to each of your reporting currencies.
Period Closing Tasks	<ul style="list-style-type: none"> <li>• Finish entering all regular and adjusting journals for the period in your ledger.</li> <li>• Post all unposted journals in your ledger if not already done in the previous step.</li> <li>• Post all unposted journals in your reporting currencies if not already done in the previous step.</li> <li>• Run Revaluation in both your ledger and reporting currencies. Post the resulting revaluation batches in each ledger.</li> <li>• As needed, translate balances in your ledger.</li> <li>• Generate needed reports from both your ledger and reporting currencies.</li> <li>• Close your accounting period in both your ledger and reporting currencies.</li> </ul>

## How Reporting Currencies Are Calculated

If you use reporting currencies at the journal or subledger level, when you create accounting, post journal entries, or translate balances, journals are posted in your reporting currency. General Ledger and Subledger Accounting automatically generate journals in your reporting currencies where the entered currency amounts are converted to the reporting currency amounts. Other factors used in the calculation of reporting currency balances are listed:

- **Manual Journals:** Enter a manual journal batch in your reporting currency at the journal or subledger level by using the Create Journals page. Select the journal or subledger level reporting currency from the ledger's list of values and continue in the same manner as entering any other manual journal.
- **Conversion Rounding:** Use the reporting currency functionality to round converted and accounted amounts using the same rounding rules used throughout your Oracle Fusion Applications. The reporting currency functionality considers several factors that are a part of the currencies predefined in your applications, including:
  - **Currency Precision:** Number of digits to the right of the decimal point used in currency transactions.
  - **Minimum Accountable Unit:** Smallest denomination used in the currency. This might not correspond to the precision.
- **Converted Journals:** Generate and post automatically, using the General Ledger Posting process, journals in your reporting currencies when you post the original journals in the source ledger for the following types of journals:
  - Manual journals

- Periodic and allocation journals
- Unposted journals from non-Oracle subledger applications
- Unposted journals from any Oracle Fusion subledger that does not support reporting currency transfer and import
- Optionally, revaluation journals
- Unconverted Journals: Rely on the subledger accounting functionality to converted and transfer Oracle Fusion subledger journals for both the original journal and the reporting currency journal to the General Ledger for import and posting. The reporting currency conversion for these journals is not performed by the General Ledger.
- Approving Journals: Use the journal approval feature to process reporting currency journals through your organization's approval hierarchy. You can enable journal approval functionality separately in your source ledger and reporting currencies.
- Document Numbers: Accept the default document numbers assigned by the General Ledger application to your journal when you enter a journal in your ledger. The converted journal in the reporting currency is assigned the same document number. However, if you enter a journal in the reporting currency, the document number assigned to the journal is determined by the reporting currency.
- Sequential Numbering: Enable sequential numbering if you want to maintain the same numbering in your reporting currency and source ledger for journals, other than those journals for Oracle Fusion subledgers. Do not create separate sequences for your reporting currencies. If you do, the sequence defined for the reporting currencies is used and can cause document numbers not to be synchronized between the ledger and reporting currencies.

---

#### Note

If the Sequential Numbering profile option is set to **Always Used** or **Partially Used** and you define an automatic document numbering sequence, General Ledger enters a document number automatically when you save your journal. If you use manual numbering, you can enter a unique document number.

- 
- Revaluation: Run periodically revaluation in your ledger and reporting currencies as necessary to satisfy the accounting regulations of the country in which your organization operates.
  - Account Inquiries: Perform inquiries in the reporting currency. Drill down to the journal detail that comprises the reporting currency balance. If the journal detail is a converted journal that was converted automatically when the original journal was posted in the source ledger, you can drill down further to see the source ledger currency journal amounts.

---

#### Note

Be careful when changing amounts in a reporting currency, since the changes are not reflected in your source ledger. Making journal entry changes to a reporting

currency makes it more difficult to reconcile your reporting currency to your source ledger. In general, enter or change your journals in your source ledger, and then allow posting to update the reporting currency.

---

---

#### Note

If you use reporting currencies at the journal or subledger level, statistical journals are generated for your reporting currencies, but the journals are not affected by the currency conversion process.

---

## Manage Journal Approval Rules

### Approving Journals: Points to Consider

Journal approval in Oracle Fusion applications uses Oracle Fusion Approvals Management (AMX) to merge the functionality of Oracle Approvals Management (AME) and PeopleSoft Approvals (AWE). In addition, Oracle Business Process Execution Language (BPEL) has replaced Oracle Workflow.

#### Rule Definition Consideration

There is one predefined approval rule. If you enable the ledger and the source for approval, then the journal entry is sent for one level of approval by default. You must configure the approval rules in the AMX Rules Setup user interface. For a simple approval scenario, start by defining one or all of the following rules.

- Journal approval based on the highest journal line amount per ledger per batch.
- Journal approval based on the highest journal amount per ledger per batch.
- Journal approval behavior based on where you are in the period close process. For example, are you in the beginning, middle, or end of the month, or in pre-close, close, post close, or quarter close process?

For example, after your ledger is enabled for approval, enter the following approval rules to apply when your maximum journal line amount is:

- Less than 50,000 United States dollars (USD), then there is no approval required
- Between 50,000 to 100,000 USD, then the journal batch requires one level of approval
- Greater than 100,000 USD, then the journal batch requires two levels of approval

Build your rules for every combination of ledger, entered amount, approval level, or other needed scenarios by using the pattern in the suggested rules. In addition, the Oracle Fusion functionality allows you to further define your own rules based on attributes from the different parts of your journal, including the

ledger, batch, header, or line level. For example, use category, source, account, or descriptive flexfield information as selection criteria for the journals to be sent for approval.

The ledger is included in the rules because you typically define approval rules per ledger. Set the options that enable journal approval at the ledger level and by journal source. This allows the approval process to determine which journals to send for approval.

### AMX List Builder Considerations

Use the following AMX List Builder to build your approval list.

List Builder	Functionality	Additional Information
Human Resources (HR) Supervisory	This method uses the HR Supervisory hierarchy levels and specifies the number of levels available for approval.	This method is most effective when the General Accountant enters the journals. For example, if an accountant enters a journal, he needs approval from his manager. If his manager enters a journal he needs approval from his manager and so on up the hierarchy for the specified number of levels. Self approval can be set at any levels in the hierarchy.
Job Level	A relative dollar amount can be attached to a job. The approval list moves up the HR Supervisory hierarchy to the point it finds a job with the necessary approval amount.	Enable self-approval to allow approval of journals created within your authority limit.
Position	A relative dollar amount can be attached to a position.	This is effective if you need a hierarchy different than the HR Supervisory hierarchy. It is also effective when there are multiple hierarchies that need to be selected based on different attributes.
Approval Group	Approver groups represent functional or subject matter experts outside the transaction's managerial chain of authority, such as Legal or HR personnel.	
Dual Chain	Dual chains can be processed at the same time.	

---

#### Note

Best practices are to select Job Level, HR Supervisory, or Position list builders for your journal approval rules.

---

#### Other Considerations

Other functionality to consider before defining approval rules include:

- Approval is for the entire journal batch regardless of the attributes used in the approval rules.
- For the job and position level approvals, the approval list continues up hierarchy until it finds the approver with the correct approval authority.
- If the journal requires approval, submitting a journal for posting automatically routes the journal for approval before posting.
- A journal can be escalated to a new approver by the administrator.
- The **Withdraw Approval** button on the Journals page is used at anytime in the approval process to withdraw journals from the process. Clicking this button allows you to edit to the journal. After your changes are made, submit the entry for approval again. When a journal is withdrawn, the completion status is set to Incomplete.
- Approval notifications display a table of key journal attributes for each journal and a list of past, current, and future approvers.
- The Journals region of the dashboard displays the journals requiring your approval (if you have the privilege to approve journals) and journals with pending approval from others.
- The Journals page allows you to approve or reject journals if you are the current approver.
- Allocation journals are not routed through the approval process.

---

**Note**

Approval is enabled at the ledger and source level. Both the ledger and journal source need to be enabled for the approval process.

---

## Manage AutoPost Criteria Sets

### Creating an AutoPost Criteria Set: Worked Example

This example shows how to create an AutoPost Criteria Set to post your general ledger journal entries that were created by the journal import process for your subledger transactions. Your enterprise, InFusion Corporation, implemented Oracle Fusion General Ledger and the following Oracle Fusion subledgers: Payables and Receivables. You use a non-Oracle subledger called Fast Assets for fixed asset tracking and depreciation. You want to automate posting of your general ledger journal batches created by the journal import process to protect the subledger sourced journal entries from edits or deletion that might inadvertently happen and cause an out-of-balance situation between your subledgers and general ledger.

Consider the following points while creating your criteria set:

- Use the **All** option for category and accounting period to reduce maintenance and ensure that all journal imports are included in the posting process.

- Create a criteria set that includes all your subledger sources. Create multiple criteria sets by source only if you need to schedule different posting times to balance close activities or reduce processing time.

## Creating an AutoPost Criteria Set

Create your AutoPost Criteria Set to automatically post journal entries from both Oracle and non-Oracle subledgers.

1. On the Manage AutoPost Criteria Sets page, click the **Create** icon to open the Create AutoPost Criteria Set page.
2. Enter the set name: All Journal Imported Entries
3. Select the **Enable** check box.
4. Enter the description: Posting journals imported from the subledgers.
5. Click the **Add Row** icon to add each new line.
6. Complete the fields, as shown in the table below:

Priority	Ledger or Ledger Set	Source	Category	Accounting Period
1	InFusion Corporation Ledger	Payables	All	All
2	InFusion Corporation Ledger	Receivables	All	All
3	InFusion Corporation Ledger	Fast Assets	All	All

7. For all three sources, select **Yes** for the **process all criteria** option and enter 30 as the number of days before and after submission date.

Setting the before and after days with a wide range of days enables the process to run less often.

8. Click the **Save and Close** button.
9. Schedule the process to run daily at 3:00 a.m.

Schedule the process immediately after the journal imports to prevent changes to the journals. Run the process during nonpeak times to save resources.

## Manually Generating the AutoPost Process: Examples

Create an AutoPost criteria set and schedule the AutoPost process to run on a regular basis following your scheduled journal imports from your subledgers. When errors occur that prevent posting of the journal imports, you must correct the errors and manually run the AutoPost process. The following scenarios illustrate the kinds of errors that could occur and how you can resolve these errors.

## Scenario

The following errors occurred and prevented the journal batches from posting when the scheduled AutoPost process ran.

Error	Cause	Solution
Error - Unopened accounting period	The journal import was imported into a future period. An error arises when the AutoPost process runs on a schedule because journals cannot be posted in a future period.	Open the period.
Error - Invalid or no journals	Journal import fails to import transactions from the general ledger interface table. The AutoPost process runs on schedule but finds no batches to post. The Posting process does not run and the AutoPost Execution report shows that no batches matched the criteria.	Correct the error that caused the journal import to fail.
Error - Invalid or no journals	No journals were selected based on the posting criteria. Journal batches are available for posting. The Posting process does not run and the AutoPost Execution report shows that no batches matched the criteria.	Revise the criteria set.

After you correct the errors, manually run the AutoPost process by selecting the **Launch AutoPost** option from the Tasks panel on the journal pages or by clicking the **Generate** button on the AutoPost criteria set pages. Verify that the process ran successfully by reviewing the AutoPost Execution report.

## FAQs for Manage AutoPost Criteria Sets

### How can I run the AutoPost process?

After you define an automatic posting criteria set, run the AutoPost process by clicking the **Generate** button on the Manage AutoPost Criteria Sets page or the **Launch AutoPost** link from the Journals task pane. The AutoPost process posts the journal batches that meet the criteria defined. Optionally, schedule the AutoPost process for specific automatic posting criteria sets through the Enterprise Scheduler to run at specific times and submission intervals.

### How can I identify errors that occurred during my AutoPost process?

Review the AutoPost process results on the AutoPost Execution report. This report is automatically created when the process completes successfully. The report contains the batch name, accounting period, and balance type for each posted journal batch, and lists error statuses for batches that failed to post. The unposted journals with their error status are also displayed on the Requiring Attention tab of the Journals work area and the General Accounting Dashboard.

## Why didn't the AutoPost process post journal batches as expected?

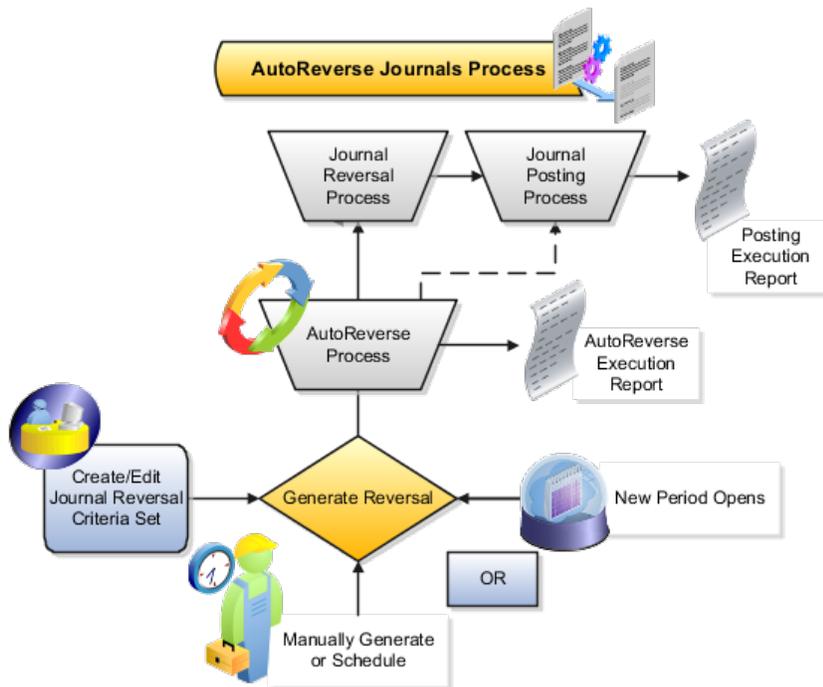
Verify that the posting criteria set specifies the precise criteria needed to post the desired journals. If the criteria is correct, then verify the following:

- Journal imports completed successfully.
- Journal batches are error free and ready to post.
- Desired accounting period is open.

## Manage Journal Reversal Criteria Sets

### Automatic Journal Reversals: How They Are Processed

The ability to submit journal reversals automatically allows you to automate and streamline your journal reversal process. If you routinely generate and post a large number of journal reversals as part of your month end closing and opening procedures, using the automatic reversal functionality saves you time and reduces entry errors.



## Settings That Affect Journal Reversals

The journal must meet the following criteria to be automatically reversed:

- Balance type is Actual.

- Category is enabled to be automatically reversed.
- Reversal period is open or future enterable.
- Posted but not yet reversed.
- Not a reversal journal. Reversal journals cannot be reversed in Oracle Fusion General Ledger.
- Not a posted journal for a reporting currency that was replicated from its source journal. Reporting currency journals that were replicated from a source journal will be reversed when the source journal is reversed.
- Not a posted journal that originated from Oracle Fusion Subledger Accounting with a frozen source.

There is a new ledger option called **Launch AutoReverse After Open Period** that you can enable to have journal reversals automatically generated when an accounting period is first opened for the ledger. This ledger option replaces the former profile option called GL: Launch AutoReverse After Open Period. If you prefer to reverse your journals on the last day of every month, disable the ledger option to automatically launch reversals when the period is opened. Then schedule the AutoReverse process to run on the last day of every month.

## How Automatic Journal Reversals Are Processed

Define Journal Reversal Criteria Sets to automatically reverse and optionally post journals using the following criteria:

Criteria	Functionality	Options
Category	Required.  The journal category you set as the reversal option. Journals entered with this category are chosen for reversal and optionally, posting.	All journal categories are listed.
Reversal period	Required.  The accounting period of the reversal journal. The <b>Next day</b> option is only applicable to average daily balance ledgers. Nonaverage daily balance ledgers and consolidation average daily balance ledgers treat the <b>Next day</b> option in the same manner as the No default option.	<ul style="list-style-type: none"> <li>• No default</li> <li>• Same period</li> <li>• Next period</li> <li>• Next nonadjusting</li> <li>• Next day</li> </ul>
Reversal day	Required for average daily balance ledgers only.  The day of the period on which to reverse the journal.	<ul style="list-style-type: none"> <li>• First day</li> <li>• Last day</li> <li>• Next day</li> </ul>
Reversal method	Required.  The method for changing the amounts in the reversal entry.	<ul style="list-style-type: none"> <li>• Change sign</li> <li>• Switch debit or credit</li> </ul>

Automatic reversal option	<p>Required.</p> <p>The option to reverse and post journals automatically. Journals are posted after they are reversed.</p>	<ul style="list-style-type: none"> <li>• None</li> <li>• Reverse automatically</li> <li>• Reverse and post automatically</li> </ul>
---------------------------	---	---

After creating your journal reversal criteria sets, assign them to ledgers. Journal reversal criteria set can be shared and assigned to multiple ledgers. Also secure journal reversal criteria set definitions using definition access set security to prevent unauthorized users from using, viewing, or modifying the journal reversal criteria.

If the automatic reversal option is set to reverse and post automatically, the AutoPost process posts all the reversal journals that were generated by the AutoReverse process. The process does not pick up other journals. You manually post reversal journals that were generated outside of the AutoReverse process.

---

**Note**

Journals posted by the AutoReverse process always bypass approval.

---

General Ledger automatically creates the AutoReverse Execution report when the AutoReverse process completes successfully. The report prints the journal name and reversal period for each journal that is successfully reversed and whether the reversal journal is submitted for posting. The AutoPost Execution report is created automatically when the AutoPost process finishes. These reports help you diagnose any problems and verify that all journals were processed properly.

---

**Note**

The AutoReverse process does not check that the reversal date is a valid business day for an average balance ledger. The journal validation in the journal pages or import process does the check and if necessary, rolls the date to the next business day.

---



---

---

## Define Period Close Components

### Manage Period Close

#### Opening First Period: Overview

For all ledgers, primary, secondary, and journal and subledger level reporting currencies, open the first period of the ledger when you are ready to transact in that period.

To open the first period of your ledgers, navigate to the Open First Period task in the primary ledger task list and click the **Go to Task** icon. On the submission page, select the ledger and the period to open. Click the **Submit** button to launch the open period process.

There are other ways to open the first period or subsequent periods without going into the Setup and Maintenance work area. You can maintain the ledgers' period statuses from the:

- Close Status region in the General Accounting Dashboard. The Close Status region provides real time visibility into the period close process from your subledgers to your General Ledger across the entire enterprise.
- Manage Accounting Periods task in the Period Close work area.
- Process Monitoring work area, which provides a framework for launching, monitoring and maintaining processes across Oracle Fusion Financials.

#### Close Monitor: Overview

The Close Monitor:

- Provides information on the period close status for a given accounting period across multiple products for related ledgers in a hierarchical ledger set based display.
- Uses the hierarchical ledger set to mirror the consolidation relationships and roll ups of entities across the enterprise.

- Summarizes period close status information for each ledger across multiple products and for each consolidation node across multiple ledgers.
- Provides the contact information of the manager for a given node on the ledger set hierarchy.
- Summarizes high level income statement results for each entity and aggregates this financial information at each consolidation node.
- Displays each of these elements of information, period status, manager information, and financial data, in separate tags that are navigated to for each node of the interactive hierarchical display.
- Provides views for a given ledger set, for a particular accounting period, and currency.

The period status information that is displayed is broken down by application module including General Ledger, Payables, Receivables, Asset, Projects, and Costing. Some modules track their entity at a more granular level, such as:

- Business units for Payables, Receivables, and Projects
- Asset Books for Assets
- Cost Organization Books for Costing

The Close Monitor indicates the number of the subunits by module for the ledgers. It also displays the fractional indicator, where applicable, of how many of the subunits are at the closed status.

Secondary ledgers, journal level, or transaction level reporting currencies cannot be associated with subledger business units for Payables, Receivables, and Projects. As such, if the ledger set displayed in the hierarchy includes members that are secondary ledgers, journal, or subledger level reporting currencies, the period status indicated in the Close Monitor for such subledger modules is based on its related primary ledger. Asset books and cost organization books can be associated with all types of ledgers. Therefore in the case of the Assets and Costing modules, their period status for secondary ledger or reporting currencies is shown accordingly for the books directly associated with them. Otherwise, their period statuses are derived from the books associated with their primary ledgers.

### **Setting Up the Close Monitor**

The Close Monitor setup is comprised of a ledger set hierarchy definition whereby a predefined ledger set is addressed, with each ledger and ledger set assigned a manager who is responsible for its financial close, and a logo to represent the entity in the display.

---

#### **Note**

The list of managers available for assignment contains the persons defined in the Human Capital Management (HCM) module of Oracle Fusion Applications. The attributes defined in HCM, such as the picture of the person and contact details, are shown in the Close Monitor.

---

The ledger set serves as the foundation of this setup.

- The members of the Close Monitor hierarchy must share a common chart of accounts and calendar.
- The financial data displayed in the Close Monitor is derived from the account group assigned to the ledger set, therefore, an assignment is required. The account group:
  - Must include two line items whose account designations respectively query the total revenues and total expenses of the organization.
  - Reflects a summarized income statement in the financial data tab of the Close Monitor.
- All ledgers in the ledger set share a common chart of accounts and the selection of accounts are equally applicable throughout the nodes in the ledger set hierarchy.
- When working with ledger sets that include members that are also ledger sets, you can choose any of the ledger sets in the selector to indicate the top starting ledger set to display in the Close Monitor.
- If different account groups are assigned to each ledger set member in such a ledger set, the account group used to display the financial data is the one assigned to the ledger set specified in the selector in the Close Monitor.
- To have meaningful comparison and summation along the ledger set hierarchy:
  - Assign ledgers to the ledger set that have a relevant currency representation that matches the intended group currency that the Close Monitor displays the financial data in.
  - Select the appropriate primary, secondary, or reporting currency ledger for assignment to the ledger set.
  - Alternately, use translated balances (balance level reporting currency) in the ledger set selection to satisfy the common group currency requirement if needed.

### **Viewing the Close Monitor**

You choose a ledger set, an accounting period, and currency as the view criteria for the Close Monitor display. You can alter this selection at any time.

For example, change the currency displayed by:

- Working with a global ledger set.
- Shifting the focus to a lower level ledger set that is aggregating at the continental level, such as North America, that uses a different group currency.
- Including the ledger with the relevant currency representation that matches the selected group currency that the Close Monitor financial data is displayed in.

---

**Note**

If matching financial data for a ledger in the selected currency is not available, a message is displayed stating that the requested financial data is not available.

---

The Close Monitor supports different zoom levels to enable you to:

- Accommodate viewing a larger ledger set hierarchy in its entirety, given the limited display area of the user interface.
- Show detail information for each node which can vary, decreasing and simplifying in content as you zoom out further to be able to accommodate showing more nodes in a single view.
- Hover over the more summarized node and view a punch out of that particular node that shows the complete set of information available at the 100% zoom level.
- Leave the zoom level at 100% and move around the display to other ledger sets or ledgers currently not in view.

---

**Note**

A view control panel that can be exposed on demand allows you to adjust the zoom level, pan across the hierarchy, flip the display tabs, and switch the hierarchy display format.

---

## Period Close Components: Explained

While implementing your accounting configuration, optionally define and maintain the period close components to customize your accounting configurations setup.

Period close components include allocations, period entries, revaluation, and historical rates.

If you use allocations, revaluation, or translation, configure the following tasks under the Define Period Close Components parent task in your implementation project:

- Manage Allocations and Period Entries
- Manage Revaluations
- Manage Historical Rates

### Manage Allocations and Period Entries

Manage Allocations and Period Entries is a manual task in the implementation project. Use the Calculation Manager to create allocations and other formulaic journal templates for generating periodic journal entries automatically. Base formulas on multiple criteria.

You must perform an external procedure outside the Setup and Maintenance work area to complete this task. In order to setup your allocations rules, navigate

to the Journals work area and click the Create Allocations Rules task from the Tasks pane. This task navigates you to Calculation Manager, a framework that enables you define your allocation rules and formulas using a graphical interface and intuitive step-by-step wizards.

### **Manage Revaluations**

Defines currency revaluation options, such as the range of accounts to revalue and the gain or loss accounts. Revaluation is done to adjust foreign entered amounts due to currency fluctuations. Navigate to the Manage Revaluations page, and define and generate your revaluation definitions.

### **Manage Historical Rates**

Historical rates are the weighted average rate for transactions that occur at different points in time. Used by the system to calculate the conversion rate on equity account balances during foreign currency translation of the balance sheet.

Navigate to the Currency Rates Manager page to define and maintain your historical rates that are used in the translation process. In Oracle Fusion General Ledger, you can currently define historical rates using an ADF Desktop Integrator spreadsheet.

To create new historical rates, specify the required Ledger and the other optional fields, as needed. Click the **Create in Spreadsheet** button to open the spreadsheet for uploading.

To update the existing historical rates for your ledgers, click the Edit in Spreadsheet button, the spreadsheet is prepopulated with the existing historical rates.

---

#### **Note**

Before using the historical rates spreadsheet, install the ADF Desktop Integrator client as an add on to Microsoft Excel.

---

## **Manage Allocations and Periodic Entries**

### **Allocation and Periodic Entries: Overview**

In Oracle Fusion General Ledger, use the Calculation Manager to create allocations and other formulaic journal templates for generating periodic journal entries automatically. Base formulas on multiple criteria. For example, use account balances or statistical amounts to allocate shared revenue or costs across multiple organizational units and ledgers. Define complex computations based on variables from different charts of accounts. Group journal formulas together and execute sequentially to update account balances in a step-by-step process.

The Calculation Manager provides flexibility, automation, intelligence, and control in distributing costs and revenues across the enterprise. In addition, the Calculation Manager:

- Distributes revenues or costs with recursive allocation rules
- Creates complex formula rules using formula components
- Contains an Allocation Wizard to define allocation and formula rules
- Uses real time checking of rule definitions to validate correctness of rules
- Minimizes setup and maintenance time with reusable components
- Simplifies allocation generation mechanism by integrating with enterprise scheduler
- Groups rules together in rule sets and cascading allocations for processing efficiencies
- Creates primary, statistical, or foreign currency allocation and formula rules

Access the Calculation Manager from the **Tasks** pane of the **General Accounting** dashboard or **Journals** work area by clicking on the:

- **Define Allocation Rules** link to define or modify allocation definitions
- **Generate Allocations** link to run the allocation process

---

#### Note

For more information, see:

- Hyperion Calculation Manager Release 11.1.2.1 Designer's Guide
- 

## Calculation Manager: Overview

The Calculation Manager creates, validates, deploys, and administers sophisticated allocation rules. In the Calculation Manager:

- Base formulas on multiple criteria, such as account balances or statistical amounts, to allocate shared revenue or costs across multiple organizational units.
- Use complex computations based on different variables to automatically calculate allocated amounts.
- Group journal formulas together and executed sequentially to update account balances step-by-step.

There are three types of objects that can be created in Calculation Manager:

- **Components:** Contain formulas, points of view, or allocation objects.
- **Rules:** Contain components such as points of view, formulas, and templates, which are used to create allocation calculations.
- **Rule Sets:** Contain sets of rules that can be calculated sequentially

---

**Note**

The following are limitation in Oracle Fusion General Ledger.

- Allocation rules cannot be shared across rule sets in Calculation Manager.
  - Within a rule or rule set, the same target or offset cannot be written to by multiple rule components.
  - When generating allocation rules with run time prompts other than the User Point of View in an allocation rule component, an error occurs.
- 

## Allocation Security: Explained

The following privileges and permissions are associated with the Calculation Manager:

- Generate General Ledger Allocation Formula - Generate Allocation and Periodic Entries: Permits generation of allocation and periodic entries.
- Define General Ledger Allocation Formula - Manage Allocation Rules or Rulesets via Calculation Manager: Grants the ability to update allocation rules or rulesets owned by the user with view access to all allocation rules or rulesets regardless of their ownership.
- Define Self Managed General Ledger Allocation Formula - Manage Allocation Rules or RuleSets via Calculation Manager: Grants the ability to update allocation rules or rulesets, but limited to the ones owned by the user.
- Administer General Ledger Allocation Formula - Administer Allocation Rules or RuleSets via Calculation Manager: Grants the ability to update all aspect of allocation rules or rulesets including the ownership attribute, regardless of the original definition's ownership.

## Calculation Manager Toolbar: Explained

In addition to the Oracle Hyperion Enterprise Performance Management Workspace buttons, the Calculation Manager toolbar displays buttons that are specific to the Calculation Manager. Not all buttons display in all the views and designers within the Calculation Manager.

The Calculation Manager toolbar consists of the following buttons:

- Home: Displays the default startup option for the content area.
- System View: Displays the main view within the Calculation Manager. (This is the default view).
- List View: Displays a list of objects that you can filter by application type, application, object or database type, deployment status, and validation status.

- Custom View: Displays a view you can customize with folders you create and objects you drag and drop into them.
- Filter Options: Opens the Filter dialog that you can use to filter objects in the List View.
- Refresh: Refreshes the view with your latest changes.

The Calculation Manager toolbar adds the following buttons when you open a rule:

- Save: Saves the object with which you are working.
- Validate: Validates the object with which you are working.
- Validate and Deploy: Validates and deploys the object with which you are working.

## Calculation Manager Menus: Explained

Calculation Manager menus and menu options display in addition to Oracle Hyperion Enterprise Performance Management Workspace menus and menu options. The menus and options vary depending on the view you are using and the object with which you are working. The default view of the Calculation Manager displays the following menus when you launch Calculation Manager, System View.

---

### Note

This topic describes the Calculation Manager menu options only.

---

### File Menu

Enables you to create new objects, open and close objects, import and export objects, print rules, and log off.

---

### Note

Not all of these file menu options are available for the products that use Calculation Manager.

---

- New, Rule: Creates a new rule
- New, Ruleset: Creates a new rule set

### Edit Menu

Enables you to edit objects you select. It is available from most of the views and from within the Rule and Component definition pages.

- Edit, Delete: Deletes an object selected in the System, List, or Custom View
- Edit, Copy: Copies selected text

- Edit, Paste: Pastes text copied to the clipboard to the right of the cursor
- Edit, Copy Group: Copies a component group

---

**Note**

The Edit menu is not available within the Deployment View.

---

**View Menu**

Enables you to open different views.

- View, View Pane: Displays or hides a list of existing and/or new objects that you can add to rules, rule sets, components, and templates by dragging and dropping them.

---

**Note**

This is the only View menu option available from within the Rule Designer and Ruleset Designer.

---

- View, List View: Displays a list of the objects you select on the Filter dialog. The filter dialog enables you to create a filtered list, by application type of applications, databases, and objects.
- View, System View: Displays a list of the Essbase applications, databases, and objects to which you have access. This is the default view.
- View, Custom View: Displays a view that you can customize with folders you create and drag and drop objects into them. This view enables you to organize objects in a way that is meaningful to you.
- View, Deployment View: Displays a list, by application type and application, of the rules and rule sets that are deployed and not deployed with their deployment and validation status.

**Tools Menu**

Enable you to install other products, search for objects, create a filtered list of objects for the List View, edit the caption of an object, and access the Variable Navigator and Variable Designer.

- Tools, Filter: Opens the Filter dialog from which you can filter by application type, application, object type (rule, rule set, formula or script component, or template), calculation type, plan type, database, deployment status, and validation status. You can also select All to display all application types, applications, objects, and databases, regardless of their deployment and validation status.
- Tools, Variables: Opens the Variable Navigator in which you can navigate to a location for which you want to create, edit, copy, or delete a variable. From the location you select in the Variable Navigator, you can display the Variable Designer in which you can create, edit, copy, and delete variables for components.

---

**Note**

The two menu options listed above are not available within the Deployment View.

---

**Actions Menu**

Enables you to validate and deploy objects you select in the views and from within the Rule and Ruleset Designers. Not all of the Actions menu options are available from within the views and designers.

- Actions, Validate: Validates the rule, rule set, and formula component you selected

---

**Note**

This is the only Actions menu option available from within the Deployment View.

---

- Actions, Deploy: Deploys the rules or rulesets you selected.
- Actions, Quick Deploy: Deploys the rule in fewer steps than regular deployment by using a shortcut to one or more applications.

---

**Note**

This feature is available only from within the Rule Designer for Essbase business rules.

---

## Using Flow Charts: Explained

View rules and templates, and the components that comprise them, in a flow chart within the Rule Designer. When you open a rule, move amongst the components that comprise it, for example, formulas, ranges, and loops, by selecting them in the flow chart. Increase or decrease the size of the flow chart to view or hide details of the components.

When you select a component in the flow chart, its properties, usages, and other information are displayed in tabs below the flow chart. As you move among the components, the tabs below the flow chart change. For example, if you open an allocation rule that contains a formula component, and select the formula component in the flow chart the following properties are displayed:

- Properties of the formula, such as name, description, application, and application type to which it belongs
- Usages of the formula, such as which rules and templates it is used in
- Text of the formula, such as the variables, members, and functions, that are displayed in the tabs below the flow chart

## Views: Explained

Views enable you to see Calculation Manager objects in different contexts. For example, the Deployment View displays objects according to whether they are deployed or not deployed. The Custom View displays objects according to filters and criteria that you select.

The Calculation Manager contains the following views:

- List View
- System View
- Custom View
- Deployment View
- View Pane

### List View

The List View contains a filtered list of Essbase applications, or databases, and objects, rule sets, rules, or formula components, according to filter criteria you specify.

### System View

The System View is the default view that is displayed when you launch the Calculation Manager. It contains a list of all of the applications and objects to which you have access. Your access privileges are determined by the role you are assigned in Shared Services. For each object, the owner, the user who made the last change, and the date the changes were last made are listed.

### Custom View

The Custom View enables you to create folders and drag and drop objects into them to create a view that contains only the objects you want. This view enables you to organize objects in a way that is meaningful to you.

### Deployment View

The Deployment View contains a list, by application type and application, of the rules and rule sets that are deployable with their deployment and validation status. From this view, select rules and rule sets in an application to make them deployable. Then deploy one or more rules or rule sets (known as a partial deployment), or you can deploy all rules and rule sets in an application (known as a full deployment).

### View Pane

The View Pane enables you to create or open an object. Display the View Pane in the left frame of the window. Depending on whether you are working in a rule or a rule set, the Rule or Ruleset Palette, is displayed in the View Pane. In the Palette, drag new and existing objects and drop them into the rule, rule set, or flow chart.

When working with views, display or hide the View Pane using the View menu. In the Custom View, drag and drop new and existing objects from the View Pane into the custom folders you create. In the System and List views, the View Pane is hidden by default. In the Deployment View, the View Pane is not available.

---

**Note**

The content of the View Pane varies depending on which view you are in and whether you are working with a rule set, a rule, a template, or a component.

---

The following table lists the tasks that can be performed from the various views in the Calculation Manager.

Tasks	List View	System View	Custom View	Deployment
Create, open, rename, delete, refresh, and close objects	Yes	Yes	Yes	Yes
Set preferences	Yes	Yes	Yes	Yes
Import and export objects	Yes	Yes	Yes	Yes
Show the usages of objects	Yes	Yes	Yes	Yes
Create a copy of objects	Yes	Yes	Yes	Yes
Print a business rule	Yes	Yes	Yes	Yes
Select views	Yes	Yes	Yes	Yes
Exit or log off Workspace	Yes	Yes	Yes	Yes
Work with favorites	Yes	Yes	Yes	Yes
Perform an advanced search	Yes	Yes	Yes	Yes
Access help	Yes	Yes	Yes	Yes
Filter objects in the view according to criteria you specify	Yes			
Work with variables	Yes	Yes		Yes
Validate objects	Yes			Yes
Create a shortcut to a business rule		Yes		
Import and export business rules and other objects		Yes	Yes	
Validate and migrate objects		Yes	Yes	
Change the owner of an object			Yes	Yes
Deploy objects				Yes

# Filtering Objects in the List View: Examples

You can use filters in the List View to filter objects according to:

- Application Type that is populated by default with the application type, such as Essbase, your application or database, in which you are creating the new rule.
- Object type, such as allocation rules, allocation rule sets, and formula components
- Deployment or validation status

## Scenario

To create a filtered list of objects in the List View:

1. From the System View, select View, List View.

The Filter dialog is displayed the first time you open the List View. If you select filtering options, then close the List View to work in the System or Custom View. When you reopen the List View, the filter dialog is not displayed. If you want to change the filtering options when you reopen the List View, select Tools, Filter to open the Filter dialog.

2. In the Filter dialog, on Filter Options, under Application Type, select Essbase.
3. Do one of these tasks:
  - For Essbase: In the Application and Object Type fields, select the applications and object types you want to display in the List View. The default is All.
  - Under Deployed Status and Validated Status, clear any check boxes of selections you do not want to display. All check boxes are selected by default.

---

## Tip

Click Reset to reset the dialog with default values.

- 
- On Advanced Options, for Object Label, select one of these options to display only objects whose names match the criteria:
    - Starts With, to display only objects whose names start with characters you specify.
    - Ends With, to display only objects whose names end with characters you specify.
    - Contains, to display only objects whose names contain characters you specify.
    - Matches, to display only objects whose names match characters you specify.

- Enter the characters that are common to the names of the objects you want to display.
- Select Ignore case, if you want to display objects whose names contain characters in either upper or lower case, even if the case does not match the case of the text you entered in step 6.
- In Created By, enter the name of the user who created objects you want to display.
- In Modified By, enter the name of the user who modified objects you want to display.
- For Created Date, select After, Before, or Between to display only objects that were created after, before, or between dates you specify. Between is the default. Click the drop down arrows to display calendars from which you can select dates.
- For Modified Date, select After, Before, or Between to display only objects that were modified after, before, or between dates you specify. (Between is the default.) Click the dropdown arrows to display calendars from which you can select dates.
- For **Any Text**, select an option to display only objects containing text that starts with, ends with, contains, or matches text that you enter. To display objects that include this text regardless of its case, select **Ignore case**.

4. Click **OK**.

## Generating Allocations and Periodic Entries Manually: Worked Example

This example demonstrates how to generate an allocation or periodic entry manually from the Oracle Fusion General Ledger.

You are the General Accountant for Infusion America Inc. You have created allocation and periodic journal entry definitions for several monthly entries. You now generate these entries.

---

### Note

Schedule allocations and periodic entries in the Journals work area for automatic generation.

---

Prior to generating the allocation and periodic entries, the following tasks must be completed:

- The period is set to **Open or Future Enterable**. You post in open periods, but generation can take place in either an open or future enterable period.
- The rules or rules sets have been defined, validated, and deployed successfully from the Calculation Manager.

- The journal balances, that are inputs for the allocation or periodic rules, are entered and posted in the proper period.

## Generating Allocations and Periodic Entries Manually

1. From the Navigator, click the **Journals** link to open the Journals work area.
2. In the task pane of the Journals page, click the **Generate Allocations** link to open the Submission page.
3. Optionally select one or all of the following options:
  - Print Output
  - E-mail me the output
  - Notify me when this process ends
4. Select a rule or rule set from the list of values.
5. Enter the submission parameters, including **Ledger, Balancing Segment Value, and Period**. The application automatically sets the last day of the submission period as the Accounting Date and Calculation Effective Date.
6. Accept the selected check box for the **Post Allocations** option to enable the process to post the journal entries.

If you deselect the check box for the Post Allocations option, you must post the entry manually or define an AutoPost Criteria Set to automatically post the journal entries.

7. Click **Submit**.

After the generation process is complete, the journal entries created by the process are available for inquiry on the Journals page.

## FAQs for Manage Allocations and Periodic Entries

### How can I access the Calculation Manager?

Login into the Oracle Fusion General Ledger application and navigate to the **Journals** work area. From the **Journals** work area select the **Create Allocation Rules** link and automatically log into the Calculation Manager in Workspace to create new allocation rules or rule sets.

---

#### Note

The application or the balances cube that is currently selected in the General Ledger Data Access Set is automatically selected in the Calculation Manager.

---

### How can I create a folder in the Custom View?

In the Custom View, create folders that contain only the allocation rules, allocation rule sets, and formulas you want to view and work with. To add objects to your folders, drag them from the Existing Objects pane and drop them into the folders.

To create a folder in the **Custom View**:

1. In the **System View**, select **View, Custom View**.
2. In the **Custom View**, right-click the **Essbase** application type, and select **New Folder**.
3. In **New Folder**, enter a name for the folder.
4. Click **OK**.

---

### Tip

You can create nested folders by right-clicking the folder you want to create a folder in and selecting **New Folder**.

---

### How can I rename a folder in the Custom View?

Rename the folders you create in the **Custom View**.

To rename a folder in the **Custom View**:

1. In the **System View**, select **View, Custom View**.
2. In the **Custom View**, expand the **Essbase** application type.
3. Right-click the folder you want to rename, and select **Rename**.
4. In the **Rename Folder**, enter a new name for the folder.
5. Click **OK**.

## Manage Revaluations

### Revaluation Process: Explained

The revaluation process is used to adjust account balances denominated in a foreign currency. Revaluation adjustments represent the difference in the value of the balance due to changes in conversion rates between the date of the original journal entry and the revaluation date. These adjustments are posted through journal entries to the underlying account with the offset posted to an unrealized gain or loss account. All debit adjustments are offset against the unrealized gain account and all credit adjustments are offset against the unrealized loss account. If the same account is specified in the **Unrealized Gain Account** and **Unrealized Loss Account** fields, the net of the adjustments is derived and posted.

For balance sheet accounts, the revaluation journal entries are reversed in the next period. **AutoReverse** can be used to automate the reversals. For income statement accounts that use the **PTD** method of revaluation, the revaluation journal entries aren't reversed since each period's revaluation adjustment is just for that period.

In Oracle Fusion General Ledger, the revaluation functionality provides the following advantages:

- Full multicurrency functionality to eliminate currency barriers across a global business
- Predefined revaluation rules to ensure consistency in generation of revaluation entries each period
- Usage of prevailing currency normalization accounting standards including:
  - US Financial Accounting Standards Board (FASB) Financial Accounting Statement No. 52 (FAS 52), Foreign Currency Translation
  - International Financial Reporting Standards (IFRS) International Accounting Standard No. 21 (IAS 21), The Effects of Changes in Foreign Exchange Rates
- Support for multiple balancing segments to provide clarity in tracking the profitability and performance for more distinct segments of the your enterprise in any currency

### Definition

When defining your revaluations, perform the following:

- Include accounts for tracking gains and losses, currency conversion rates, and the number of transaction currencies to revalue.
- Define separate revaluation definitions for each class of accounts, using a different rate type for each class.
- Choose various conversion types and methodologies for different account ranges, such as current rates and year-to-date (YTD) method for balance sheet accounts, and average rates and period-to-date (PTD) method for income statement accounts.

---

### Note

Income statement accounts can also be revalued using YTD method.

---

Hierarchies and flexible account selection criteria, such as usage of parent values from your account hierarchy, streamlines maintenance of revaluation definitions. The parent values can be selected for the primary balancing and the natural account segments using the **Is a last descendant of** operator. Leveraging hierarchy versions extends your revaluation definitions during organizational changes. Adjust account selection criteria monthly to retrieve the accounts that need to be revalued for the current accounting period.

Share revaluation definitions across ledgers that have the same chart of accounts to reduce maintenance.

### Generation

Generating revaluations include:

- Using defined revaluation criteria and automatically generating entries to shorten your close process.

- Selecting automatic posting as part of the generate revaluation criteria to help you to achieve processing efficiency.
- Scheduling revaluations to run during off peak hours to save your system resources.
- Utilizing date effective account hierarchies to generate revaluations to keep results in line with your current organization structures.

Always run revaluation to bring monetary balances to current rates before performing currency translation or remeasurement.

### **Revaluation Execution Report**

The Revalue Balances process automatically generates the Revaluation Execution report when you run revaluation. This report shows the details of your account balance revaluation and the journal batches created after running revaluation. The report includes the currencies and revaluation rates used to revalue your accounts, the unrealized gain or loss account in which you recorded net gains and losses, and the range of accounts revalued. The report also prints the names of your batch and journals that the revaluation process creates for each foreign currency, as well as the total debits and credits of the created entries.

If the Revaluation process cannot locate rates for one or more currencies, balances are not revalued for those currencies. In this case, the Revaluation process completes with a warning and the execution report lists which currencies are missing rates.

## **Accounting for Unrealized Gain or Loss on Revaluation: Explained**

Revaluation launches a process that revalues the ledger currency equivalent balances for the accounts and currencies you select, using the appropriate current rate for each currency. Resulting unrealized gain or loss amounts are posted to the unrealized gain or loss accounts or to the cumulative translation adjustment (CTA) account you specify, and are balanced by balancing segment values. This process creates a revaluation journal which can be posted automatically.

Oracle Fusion General Ledger creates journal entries to adjust the ledger currency balances for conversion rate fluctuations, in accordance with Statement of Financial Accounting Standards (SFAS) No. 52, Foreign Currency Translation and International Accounting Standard (IAS) 21, The Effects of Changes in Foreign Exchange Rates.

The revaluation journal entries generated and posted in the primary ledger are automatically generated, converted, and posted to each of their reporting currencies. Define the CTA account for unrealized gains or losses in the reporting currency prior to running revaluation.

## **Income Statement Accounts Revaluation Rule: Explained**

Revaluation is the process which adjusts asset or liability accounts that may be materially understated or overstated due to a fluctuation in the conversion rate

between the time the transaction was entered and the time revaluation takes place. You may want to revalue income statement accounts as well. The Income Statement Accounts Rule indicates whether period-to-date (PTD) or year-to-date (YTD) method is to be used when revaluing income statement accounts.

Click the **Income Statement** radio buttons on the **Create Revaluation** page to specify whether you want to revalue income statement accounts using PTD or YTD balances. There are two radio buttons, one for PTD and one for YTD.

If you select to revalue PTD balances for income statement accounts, the process continues to appropriately revalue YTD balances for balance sheet accounts. In the revaluation definition if the range of accounts consists of both income statement and balance sheet accounts and you select PTD as an option for income statement account revaluation rule, a separate revaluation journal is created for the income statement accounts. Revaluing the PTD balance of your income statement accounts creates weighted average YTD balances using period rates from each corresponding period against the PTD account balance in compliance with the Statement of Financial Accounting Standards (SFAS) No. 52, Foreign Currency Translation.

To summarize, when you run revaluation on your income statement accounts, the process produces two separate journal entries; one that revalues your balance sheet accounts and another for your income statement accounts. You do not need to reverse the PTD revaluation journal entry for your income statement accounts in the subsequent period since that revaluation only applies to last period's activity.

---

**Note**

This functionality only applies when the range of accounts to be revalued in the revaluation definition consist of income statement accounts in addition to balance sheet accounts. Normally only balance sheets accounts are revalued.

---

## Revaluing Across Multiple Balancing Segments: Worked Example

This example demonstrates how to revalue foreign currency balances across multiple balancing segments. Your company, InFusion America, Inc. has three lines of business. You revalue your foreign currency account balances for two of your divisions, Air Components and Repair Parts. Your Installation Services line of business does not have foreign currency transactions. Your company is your primary balancing segment and your lines of business are represented in your secondary balancing segment.

---

**Note**

Enable up to three balancing segments to use the multiple balancing segment feature.

---

The following are points to consider in running the revaluation process.

- Revaluation posts the resulting gain or loss amounts against the unrealized gain or loss accounts, substituting the balancing segment values appropriately for all balancing segments.

- Gain or loss accounts and revaluation account ranges are not validated against your data access set security when the revaluation definition is created because the ledger context is not known at the time of definition.
- Data access set security is enforced when the Revalue Balances process is executed. Limited write access to the gain or loss accounts due to inadequate access results in an error.
- Segment value security rules are enforced when you enter the account ranges and the unrealized gain and loss accounts. Only segment values you have access to are available in the list of values.
- Account ranges you have read and write access to are revalued. Account combinations that you do not have access to are ignored.
- Revaluation expands the parent primary balancing segment to the child values. Data access set security applies to the child values only, not the parent value.
- Posting supports multiple balancing segments for calculating the entry to the Cumulative Translation Adjustment accounts when replicating revaluation journals to reporting currencies.

## Defining Revaluations

1. From the **Manage Revaluations** page, click the **Create** icon.
2. Enter the values in the following table in the correct fields.

Field	Value
Name	InFusion America Revaluation
Description	Revaluation for all foreign currency balances.
Chart of Accounts	InFusion America Chart of Accounts
Currency	Leave blank <hr/> <b>Note</b> If left blank, all currencies are revalued and after saving, the field automatically displays: All currencies. <hr/>
Conversion Rate Type	Daily
Days to Roll Forward	5
Unrealized Gain Account	011-00-96600000-0000-000-000
Unrealized Loss Account	011-00-96700000-0000-000-000
Income Statement Account Basis	PTD
Post Automatically	Yes

3. In the **Revaluation Accounts** region, click the **Add Row** icon.
4. Click the **Change filter conditions** icon to enter the filter used to select the accounts to revalue.

- Click the **Add Field** drop down arrow and select your company, InFusion America Inc. from the list.

Field	Value
Equals	011

- Click the **Add Field** drop down arrow and select your two Lines of Business: 30 for Air Components and 40 for Repair Parts.

Note: Your Installation Services line of business, 50, is not included because it does not have foreign currency transactions.

Field	Value
Between	30
	40

- Click the **Add Field** drop down arrow and select Account from the list.

Field	Value
Between	10000000
	29999999

- Click OK to accept your filters.
- Click the **Save and Close** button to save your revaluation.

Optionally, select the **Save** and **Generate** buttons to run the revaluation immediately.



---

---

## Define Applications Coexistence Configuration

### Integration with Oracle E-Business Suite and Oracle PeopleSoft: Overview

Oracle Fusion Applications provides a coexistence strategy that allows you to continue to use your Oracle E-Business Suite or Oracle PeopleSoft General Ledgers and subledgers while using Oracle Fusion Accounting Hub for financial reporting. Coexistence includes the ability to transfer balances from the Oracle E-Business Suite General Ledger and journal entries from Oracle PeopleSoft General Ledger to the Oracle Fusion Accounting Hub.

For more information on completing the post-installation setup for coexistence with Oracle E-Business Suite General Ledger see:

- **Configuring Oracle Golden Gate to Integrate the E-Business Suite Ledger with Fusion Accounting Hub** on My Oracle Support
- Oracle Fusion Accounting Hub Implementation Guide
- Oracle General Ledger Implementation Guide Release 12.2: See this guide for information on loading and transferring data from Oracle E-Business Suite to the Oracle Fusion Accounting Hub.

For more information on completing the post-installation setup for coexistence with Oracle PeopleSoft General Ledger, see:

- Oracle Fusion Accounting Hub Implementation Guide
- PeopleSoft General Ledger 9.1 Documentation Update: Integrating PeopleSoft General Ledger with Oracle Fusion Accounting Hub
- PeopleSoft General Ledger 9.1 Integration to Oracle Fusion Accounting Hub Implementation Guide

---

#### Note

The Oracle Data Integrator (ODI) component (extract file for manual import) is currently available via My Oracle Support only in note id: 1365971.1.

---

## Register Applications Coexistence Instances

Register applications coexistence instances to indicate in Oracle Fusion General Ledger which Oracle E-Business Suite and Oracle PeopleSoft instances are integrated with the Oracle Fusion Accounting Hub. There is a user interface to this registration. For each E-Business Suite or PeopleSoft instance, provide a unique system identifier. This identifier must also be registered in the corresponding Oracle E-Business Suite or Oracle PeopleSoft instance.

You can specify a unique journal source per instance. For Oracle E-Business Suite, you can limit which instance and balancing segments may post to a particular Oracle Fusion General Ledger.

For Oracle E-Business Suite, determine the Function ID to move data from Oracle Fusion General Ledger to Oracle E-Business Suite General Ledger. You must include the Function ID at the end of the drill down URL that is provided during the registration of the Oracle E-Business Suite instance.

To find the Oracle E-Business Suite Function ID:

1. Login as a System Administrator and navigate to Function page
2. Query for the function name: GL\_FUSION\_EBS\_DRILL
3. Go to the Help menu, click **Diagnostics > Examine**
4. Select the **FUNCTION\_ID** field. The value box shows the value of the Function ID.

For Oracle E-Business Suite: The URL format for the non-dynamic portion needs to be in the following format: `http://<domain>:<port>/OAA_HTML/RF.jsp?function_id=<function_id>`.

In the above URL format, the domain, port, and function\_id are for the Oracle E-Business Suite Instance.

For Oracle PeopleSoft: The URL format for the non-dynamic portion needs to be in the following format: `http://server/servlet_name/SiteName/PortalName/NodeName/c/PROCESS_JOURNALS.FUS_DRILLBACK_JRNL.GBL`

In the above URL format:

- `http://server/`: Scheme (http or https) and the web server name.
- `servlet_name/`: Name of the physical servlet that the web server invokes to handle the request.
- `SiteName/`: Site name specified during Oracle PeopleSoft Pure Internet Architecture setup.
- `PortalName/`: Name of the portal to use for this request.
- `NodeName/`: Name of the node that contains the content for this request.

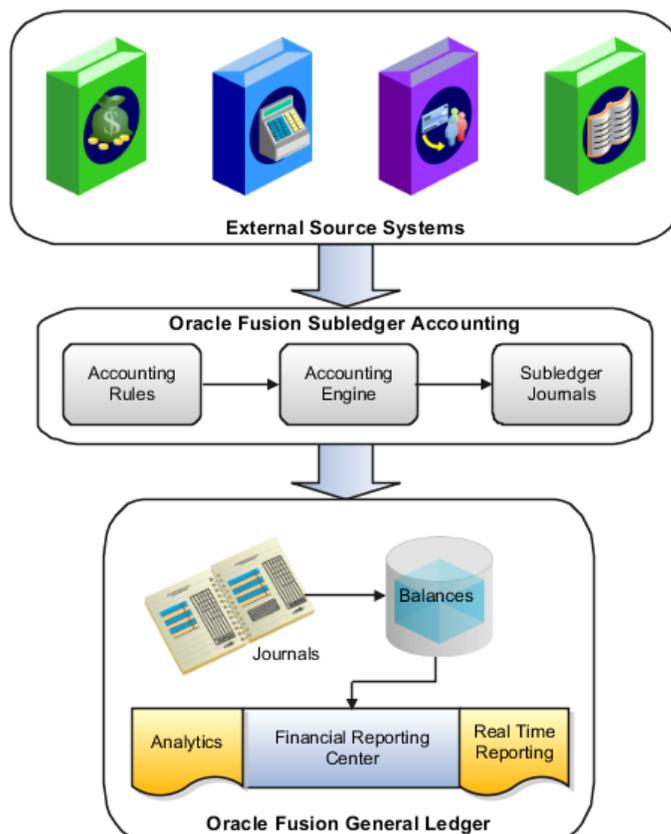
---

## Define Accounting Transformation

### Accounting Transformation Configuration: Overview

### Accounting Transformations: Overview

Oracle Fusion Accounting Hub creates detailed, auditable journal entries for source system transactions. The subledger journal entries are transferred to the Oracle Fusion General Ledger. These general ledger journals are posted and update the general ledger balances. Then the balances are used by the Financial Reporting Center for reporting and analysis. The following figure depicts this process.



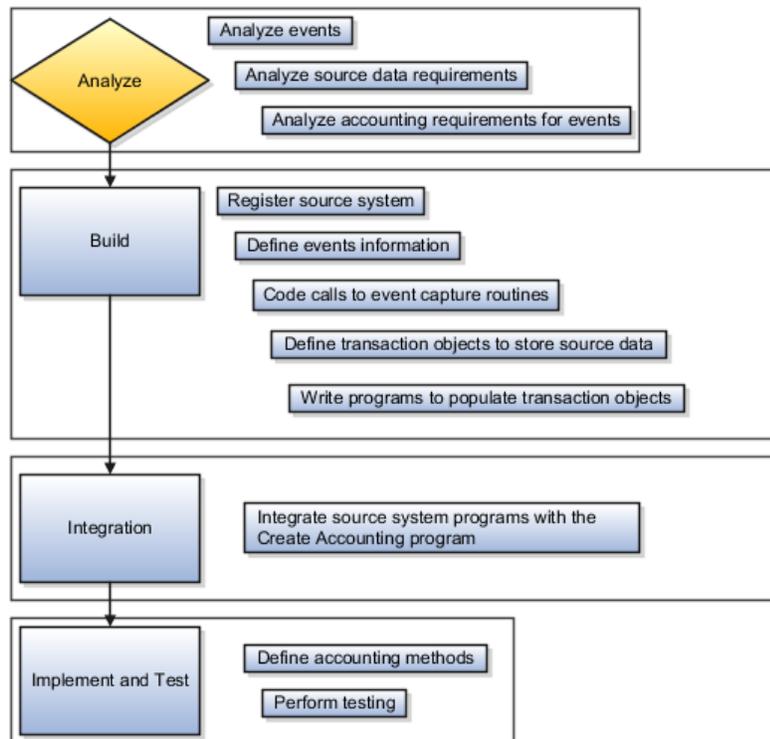
Accounting transformations refer to the process of converting transactions or activities, referred to as accounting events, from source systems into journal entries. Source systems may be diverse applications that have been purchased from non-Oracle software providers or created internally. Often, source systems are industry specific solutions. Examples of source systems are core banking applications, insurance policy administration systems, telecommunications billing systems, and point of sales systems.

### Accounting Transformation Flow Chart

When using the accounting transformation implementation process:

- Start with an analysis of your current system.
- Determine which source systems have transactions or activities that need to be accounted and reported via the Oracle Fusion Accounting Hub.
- Register the source system transactions and activities which have financial impact in the Accounting Hub to make them eligible for accounting.
- Create accounting rules that indicate how each of the accounting events is accounted.
- Group these rules together and assign them to ledgers to create a complete definition of the accounting treatment for the transactions and activities from the source system.

The following figure summarizes the accounting transformation process.



### Accounting Transformation Steps

Complete the steps described in the following table in the order listed to account for accounting events of each of your source systems.

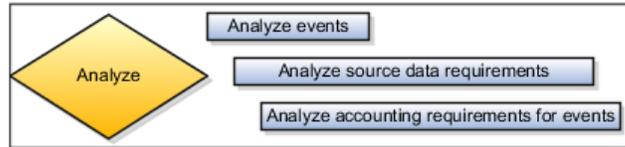
Implementation Phase	Step Number	Description
Analysis	1	Analyze source system transactions or activities to determine what accounting events to capture.
	2	Analyze transaction objects requirements. <ul style="list-style-type: none"> <li>• What source transaction information is available and needed for accounting?</li> <li>• What reference information is needed for reconciliation and reporting?</li> </ul>
	3	Analyze and map the source system's current accounting.
Definition and Build	4	Register source systems and define event model, including: process categories, event classes, and event types.
	5	Code calls to event capture routines.
	6	Build programs to extract the information from the source systems and populate it in the Accounting Hub transaction objects.
	7	Run the Create and Assign Sources program and revise source definitions and map accounting attributes.
Integration	8	Create programs that capture accounting events and their related information and send it to the Accounting Hub.
Implement and Test	9	Create accounting rules.
	10	Perform comprehensive testing to ensure that all accounting is correctly generated.

## Accounting Transformations Analysis: Explained

The analysis phase of accounting transformation implementation includes three steps.

The steps are:

- Analyze Accounting Events
- Analyze Source Data Requirements
- Analyze Accounting Requirements for Events



## Analyze Accounting Events

Some business events have financial accounting significance and require the recording of financial information. These business events are known as accounting events and provide the data used in accounting transformations.

Examples of business events from a revenue recognition or billing system include:

- Complete an invoice
- Record a payment
- Record late charges

Examples of business events from a point of sale system include:

- Record an order
- Accept a payment
- Receive a return

Examples of business events from a loan (core banking) system include:

- Originate a loan
- Fund a loan
- Record late charges for a loan

An accounting event and its associated transaction data typically relate to a single document or transaction. However, the nature of source systems may prevent them from extracting this discrete information and sending it to Oracle Fusion Accounting Hub for processing. In some cases, summarized event information, such as overall customer activity for the day, is sent for accounting transformation.

The first task is to carry out a complete analysis to determine which accounting events are captured. This analysis incorporates both the functional requirements for accounting for the source system events, as well as a review of how the events can be captured. There may be limitations on the source system, as well as volume considerations that make it desirable to capture summarized event information such as total customer activity for a day.

Complete the following analysis to identify accounting events:

- Identify the life cycle of documents or business objects and the transactions that affect their status.
- Identify events in the life cycle that may have financial implications to ensure that they are captured and accounted.

- Identify all business events for which contextual data and transaction amounts are available.

This is not a mandatory requirement, but it provides maximum flexibility for defining accounting transformation rules.

### **Analyze Source Data Requirements**

Verify that all the required sources, such as accounting amount and date, that can potentially be used to create subledger journal entries are included in the accounting transaction objects. Sources are the appropriate contextual and reference data of transactions. They provide the information that is used to create subledger journal entries. For example, the following items could be used:

- Transaction amount as the debit amount
- Transaction date as the accounting date
- Customer account ID as part of the description

Complete the analysis to determine what source data is necessary to successfully create subledger journal entries from transactions.

Flexibility in creating accounting rules is dependent on the number of sources available. There is a balance between providing all the information that can be extracted versus how much is practical to send based on your processing resources. The following list provides examples of source data:

- Amounts including entered, accounted, and conversion data
- Dates
- Descriptions
- Accounts
- Customer information
- Transaction type information

Study the transaction objects data model used by the Oracle Fusion Accounting Hub. The data model provides detailed information about the different types of transaction objects. Transaction objects are the views and tables that store transaction data in the standardized form required by the Create Accounting process.

When specifying optional header and line objects, use single table views. If you specify optional objects as multi-table views, it can result in poor performance.

Data stored in transaction objects must also satisfy accounting transaction objects validation rules. These rules verify both completion and validity of the data.

### **Analyze Accounting Requirements for Events**

Some source systems may already produce accounting entries, while others may produce raw transactions with no associated accounting. As part of the analysis, determine how much transformation is required to produce subledger journal entries. Once this is done, examine the subcomponents of the journal entry

rule set to determine how to complete rules to produce the required subledger journal entries. This exercise helps determine which subledger journal entry rule set subcomponents must be defined for the source systems data to be properly transformed into subledger journal entries.

Journal entry rule set subcomponents include the description rule, account rules, journal line rules, and supporting references.

Such an analysis should, at a minimum, answer the following questions:

- Under what conditions is each of the lines in the subledger journal entry created?
- What is the line type, debit or credit, of each subledger journal entry line?
- What description is used for the subledger journal entry?
- How are the accounts derived for the entry?
- What information may be useful for reconciling the subledger journal entry to the source system?

---

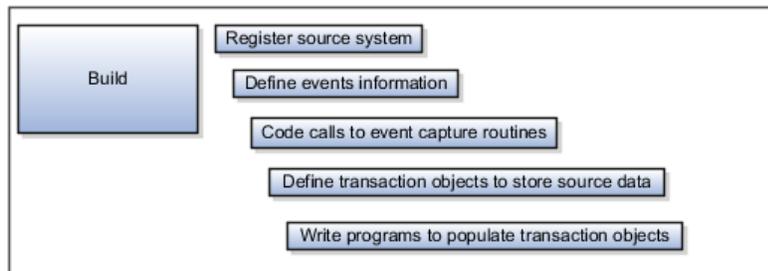
#### Note

This list is not comprehensive.

---

## Accounting Transformations Definition and Build: Explained

This section describes the steps for the accounting transformation definition and build phase of the Oracle Fusion Accounting Hub implementation.



### Register the Source System and Define Events Information

After registering the source system, set up the accounting event model. The accounting events from the source system are registered in the Oracle Fusion Accounting Hub.

#### Define Process Categories

- Process categories group event classes, and can be used to restrict the events selected for accounting when users submit the Create Accounting process. Selecting a process category when submitting the Create Accounting process indicates that all associated event classes and their assigned event types are selected for processing. This may be useful for segmenting events due to processing volumes.

## Define Event Classes

- Event classes represent transaction types and are used to group event types. For example, when accounting for transactions from a banking system, group the event types Loan Origination, Loan Scheduled Payment, and Loan Late Payments into an accounting event class for Loans.
- For each event class, register the transaction objects that will hold source data for the event types in that class.

## Define Event Types

- For each transaction type specify each business event that can have an accounting impact.

## Code Calls to the Accounting Event Capture Routines

Using application programming interfaces (APIs), create programs to capture the accounting events. The Create Accounting process combines the event information with the transaction object information and the accounting rules to create subledger journal entries. The Create Accounting process reads the event type for each event. Based upon the event type and the primary ledger, it determines which set of accounting rules should be applied to create the subledger journal entry. Once it determines which rules to use, it gets the information from the event and the transaction object rows related to the event to create the journal entry.

The following APIs for creating and updating accounting events are provided:

- Get Event Information APIs to get event information related to a document or a specific event
- Create Event APIs to create accounting events, individually or in bulk
- Update Event APIs to update events and keep them consistent with related transaction data
- Delete Event APIs to delete events

## Define Transaction Objects and Write Programs

Transaction objects are tables or views defined for each event class, capturing source transaction data for accounting events. The Create Accounting process gets the source transaction data from the transaction objects to generate journal entries.

To build transaction objects, perform following tasks:

- Create and register transaction objects under `ORACLE_FUSION` schema that are used to store the source transaction data.
- Write programs that populate the transaction objects with source values for each accounting event.

There are different types of transaction objects, indicating whether they are used at the header or line level, and whether they hold translated values:

- Transaction object headers for header level sources that are not translated
- Transaction object headers Multi Language Support (MLS) for translated header level sources
- Transaction object lines for line level source values that are not translated
- Transaction object lines MLS for translated line level sources

Header sources have the same value for all transaction lines or distributions associated with an accounting event. These sources are typically associated with a transaction header or with transaction reference data. An example of a header standard source for a mortgage loan is the loan number. A mortgage loan can have only one loan number. This number would be on the header transaction object and would not vary by line number.

Line sources have values that can vary by the transaction lines or distributions associated with an accounting event. They must be stored in the transaction objects at the line level.

Transaction objects can be mandatory or optional. At least one header transaction object is mandatory.

When creating optional transaction objects, specify them as single table views. Specifying optional objects as multi-table views may result in poor performance.

It is also possible for accounting event classes to share transaction objects. For example, when accounting for a core banking system, use the same transaction objects line table or view for both of the event classes: Fixed Rate Mortgages and Variable Rate Mortgages.

Transaction objects need to be populated before the accounting for the events occurs; otherwise, the source transaction information will not be available to generate the journal entries for the events.

Transaction objects can be populated in advance of running the Create Accounting process or they can be populated as part of the Create Accounting process by customizing `xla_acct_hooks_pkg` to automate this coordination.

## Accessing the Transaction Objects

Transaction objects should be created under the `FUSION` schema. Then, select, insert, update and delete access should be granted to `FUSION_RUNTIME` for all the transaction view and objects.

A sample grant command is: `Grant SELECT, INSERT, UPDATE, DELETE ON FUSION.XXFAH_TRX_HEADERS_V TO FUSION_RUNTIME; if XXFAH_TRX_HEADERS_V is created as a transaction object.`

## Create Sources

After the transaction objects are registered, sources and source assignments to event classes are created based on these objects. Assigning sources to event classes makes them available for creating accounting rules for those classes. The transaction objects column names are used to generate sources. Each column in each transaction object is registered as a separate source. These sources are used in the definition of accounting rules used in creating journal entries.

Create and Assign Sources process also validates the transaction objects by verifying that.

- All transaction objects, defined for the event class, exist in the database.
- All transaction objects, based on the transaction object type, contain the appropriate primary key columns of the correct data type.
- The syntax of all join conditions between the transaction and reference objects is correct.
- A reference object is not registered multiple times for an event class.
- A reference object is not assigned to another reference object.
- Existing sources or source assignments created from previous run of this process continue to be consistent with the transaction objects.
- Any existing accounting attributes mapping previously done continue to be consistent with the transaction objects and accounting attribute definitions.

### **Revise Source Definitions and Assign Accounting Attributes**

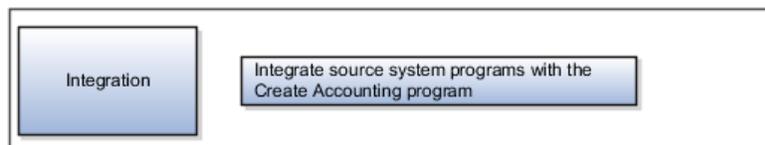
Once sources have been created, revise the source definitions before they can be used. These revisions are:

- Source names are the same as the transaction object column names. These can be revised to be more business user friendly so they are easily understood when configuring accounting rules.
- Sources that correspond to accounting flexfield identifiers are marked as accounting flexfield.
- Whenever appropriate, sources can have lookup types or value sets assigned. Assigning a value set or lookup type allows you to predefine valid values for the source that is used to create accounting rules.

After the sources are created, they need to be mapped to the accounting attributes for each event class. An accounting attribute is a piece of the journal entry; the mapping of sources to accounting attributes specifies how the Create Accounting process gets the value for each piece of the journal entry. For example, an attribute of entered currency is used to map source values to the entered currency field for subledger journal entry lines.

## **Accounting Transformations Create Accounting Process: Explained**

This section describes the accounting transformation implementation steps for the integration of source system programs with the Create Accounting process in the Oracle Fusion Accounting Hub.



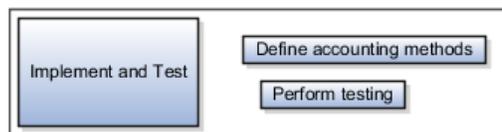
## Integrate Source System Programs with the Create Accounting Process

For Oracle Fusion Subledger Application, integrate source system programs to create accounting events using application programming interfaces (APIs).

For Oracle Fusion Accounting Hub, you need to customize the `xla_acct_hooks_pkg`.

## Accounting Transformations Implement and Test: Explained

This section describes the steps in the accounting transformations implement and test phase of the Oracle Fusion Accounting Hub implementation.



Accounting methods group subledger journal entry rule sets together to define a consistent accounting treatment for each of the accounting event classes and accounting event types for all source systems.

The following steps must be completed:

- Define accounting methods
- Perform testing

### Define Accounting Methods

Define accounting methods to group subledger journal entry rule sets to determine how the source system transactions are accounted for a given ledger. Your goals in defining accounting methods are to:

- Ensure regulatory compliance
- Facilitate corporate financial reporting
- Enable audits
- Facilitate reconciliation to source systems

Assign journal entry rule sets to event class and event type combinations in an accounting method to determine how the subledger journal entries for that class or type are created.

The following are the subcomponents of a journal entry rule set:

- **Journal Line Rules:** Determine basic information about a subledger journal entry line. Such information includes whether the line is a debit or credit, the accounting class, and if matching lines are merged.
- **Description Rules:** Determine the descriptions that are included on subledger journal entry headers and lines. Include constant and source values in descriptions.

- **Account Rules:** Determine which account should be used for a subledger journal entry line.
- **Supporting References:** Optionally used to store additional source information about a subledger journal entry and are useful for reconciliation of accounting to source systems.

You can attach conditions to journal line rules, description rules, and account rules components. A condition combines constants, source values, and operands to indicate when a particular journal line rule, description, or account rule is used. For example, for mortgage loans, you can elect to use a specific loan receivable account based on the loan type.

## Perform Testing

Once the setup is complete, testing should be comprehensive to ensure that all accounting is correctly generated. To complete testing, use accounting events and information from the source system, that is populate the transaction objects. This should, at a minimum, include testing that:

- Accounting events are successfully created.
- Sources are available for creating subledger journal entries.
- Subledger journal entries accurately reflect the accounting rules. Test that:
  - Subledger journal entries contain the appropriate dates, amounts, descriptions, and accounts.
  - Conditions used to determine journal line rules, account rules, or descriptions are valid.
- Entries are successfully summarized when transferred to the general ledger.
- Subledger journal entries are successfully transferred and posted to general ledger.

---

### Note

The above list is not intended to be comprehensive.

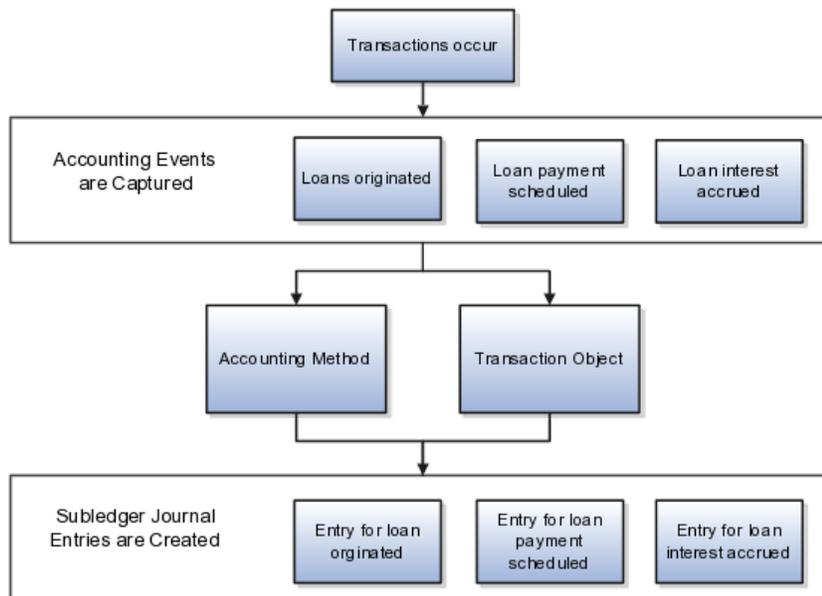
---

## Analyze Accounting Events

### Accounting Event Analysis: Overview

Accounting events have financial accounting significance and are used as a basis for recording financial information.

The diagram below describes the process to create subledger journal entries from accounting events using a custom Loans application as an example and is explained in the succeeding text.



As illustrated in the above diagram, after transactions occur, accounting events are captured to record their accounting impact. Accounting events can be captured as transactions are generated in the source system, or they can be captured in bulk as part of a standard daily close or batch process. When accounting events are captured is based upon the flexibility of the source system and the frequency with which accounting should be created. For each eligible event, the transaction object contains contextual information about the event, such as source values.

The setups associated with a ledger use the source data for the event to create the appropriate subledger journal entry. Each accounting event can be used to create one or more subledger journal entries. Subsequently, the accounting event links transactions to their corresponding subledger journal entries.

## Capturing Accounting Events: Explained

All business events that can have an accounting impact should be captured as accounting events.

### Identification and Significance of Accounting Events

The following procedures can assist in the analysis and identification of accounting events:

- Identify transactions that may have a financial impact on your organization.
- Identify the life cycles of these transactions and the business events that change the state of the transactions throughout their life cycles.

Business events vary by industry and organization. Examples of business events include the contract, order and delivery of goods and services, and receipts and payments to third parties.

Some business events have financial significance. Their impact must be accounted.

The following examples have a financial accounting impact and therefore are accounting events:

- Originating a loan
- Funding a loan
- Applying payment to a loan

Not all events are accounted. As an example, consider a loan application. The loan origination event results in accounting if the accounting method is Accrual, but may not result in accounting if the accounting method is Cash Basis.

## Accounting Event Attributes: Explained

When an accounting event is captured, different event attributes are passed to the Oracle Fusion Accounting Hub.

The following table lists some of the attributes that are stored for an accounting event along with their corresponding descriptions.

Attribute	Description
Event ID	Accounting event internal identifier; provided by Create Accounting.
Event Number	Unique event sequence number within a document. Create Accounting processes events in the order of their event number.
Event Type	Represents the business operation that can be performed on the end user transaction type event class and has accounting significance.
Transaction Identifiers	Identifies the document or transaction in the subledger tables and constitute the primary key of the transaction.
Event Status	<p>Available statuses are:</p> <ul style="list-style-type: none"> <li>• Incomplete The application is not yet ready to account for the event.</li> <li>• Unprocessed Finalized accounting has not yet been generated for this event.</li> <li>• No Action There is no accounting that needs to be done for this event.</li> <li>• Processed Accounting was done for this event in final mode with no validation failures.</li> </ul>

Event Date	Date of the accounting event that originated the journal entry.
Transaction Context Values	<ul style="list-style-type: none"> <li>• Legal Entity</li> <li>• Ledger</li> <li>• Asset Book</li> </ul>
Application Specific Attributes	Additional columns are provided for implementers to store data drawn from the transaction model (state) at the time the event is captured. These can be useful in cases where the transaction data changes and information is needed on the original event.
Security Context Values	Provide the event's security context. Examples include organization identifier and asset book.
On Hold Flag	<p>Indicates whether there is a gap ahead of an accounting event. If there is a gap, the event is put on hold.</p> <p>The Oracle Fusion Accounting Hub does not process accounting events on hold due to a gap.</p>

The Oracle Fusion Accounting Hub event tables store the event data for these attributes. The presence of this data enables the creation of individual subledger journal entries for each accounting event. It also provides an audit trail between the subledger journal entry and the transaction data of the accounting event.

Event capture routines populate these tables when the events are created.

## Transaction Context Values: Explained

For the accurate creation of subledger journal entries, an accounting event's transaction context must be passed to the Oracle Fusion Accounting Hub. This context includes the following:

### Ledger (Mandatory)

Each accounting setup requires a primary ledger that acts as the main record keeping ledger for one or one or more legal entities that use the main chart of accounts, accounting calendar, subledger accounting method, and currency to record and report on all of financial transactions. To maintain an additional accounting representation, use secondary ledgers.

### Valuation Method (Optional)

In some cases, the rules for how accounting events should be accounted require that they be subject to differing monetary valuations (valuation methods) for different ledgers. The Accounting Hub assumes that each subledger transaction and accounting event is for a single valuation method.

### Legal Entity (Optional)

Accounting events inherit the legal entity from the transaction it is associated with. Each transaction entity is stamped with a single legal entity.

## Event Status: Explained

The event status is an indicator of what actions have been completed on the transaction and what operations are yet to be done.

The Create Accounting process makes updates to this status as the accounting process progresses. Once Create Accounting successfully processes the accounting event, the status of the event will be updated to Processed.

### Event Status Details

The table below lists the event statuses along with their corresponding details. At any point of time, an event can have only one of these statuses.

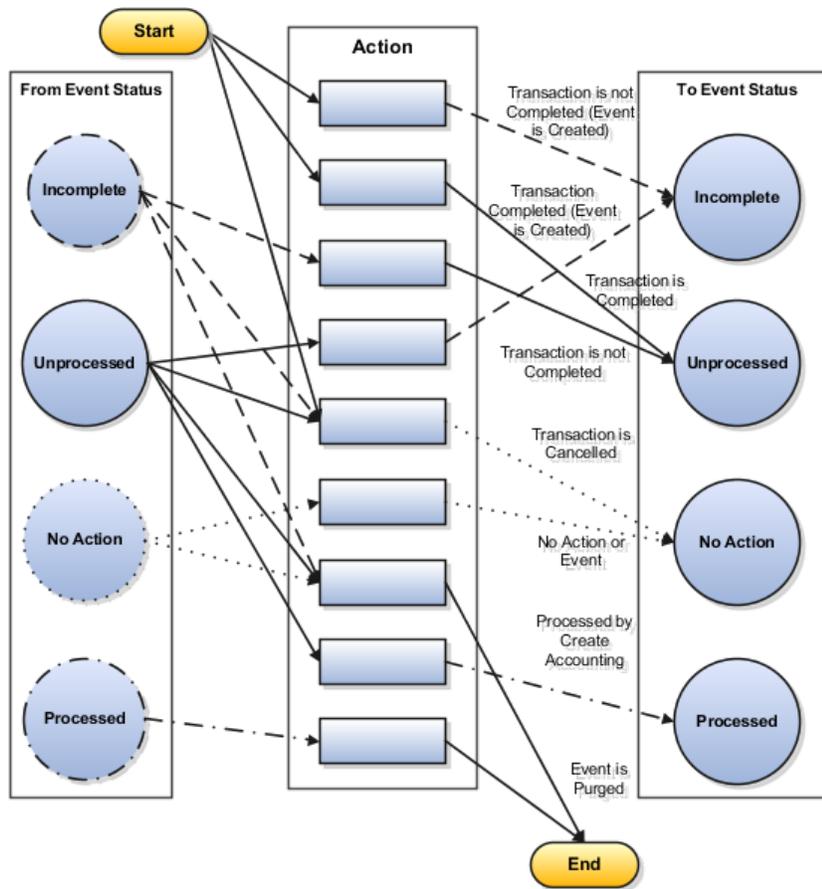
Status	Details
Incomplete	The accounting event data is in the process of being created. Some of the accounting event data cannot be created at this point. There can be validations that have not yet been performed. No subledger journal entry exists for the accounting event. If Create Accounting is run, it does not process accounting events with a status of Incomplete. The subledger application updates this event status.
Unprocessed	All of the transaction data for this accounting event has been created and all validations have been performed. At this point of time, the subledger journal entry can be created. When Create Accounting is run, it processes accounting events with a status of unprocessed.
No action	This status is set when creating or updating events using APIs. No subledger journal entry is needed for this accounting event.
Processed	<p>All of the transaction data for this accounting event is created, all validations are performed and, if appropriate, the final subledger journal entry is created. The transaction data associated with the accounting event should not be changed.</p> <p>For those transactions where multiple accounting events are allowed, any changes to the transaction data at this point of time results in a new accounting event. The changed transaction data is tracked under the new accounting event. After successfully creating subledger journal entries, Create Accounting updates the event status.</p> <p>However, you can enter new lines provided the subledger functionality allows such a change. New lines entered are recorded with new accounting events.</p>

### Event Status and the Accounting Event Life Cycle

Every event has a life cycle. The event status indicates what actions have been completed on a event. An accounting event does not necessarily go through each of the statuses.

## Accounting Events Life Cycle

Possible event statuses are displayed below.



The above diagram illustrates all the possible status changes for accounting events. An accounting event will not necessarily go through each of the statuses.

The diagram has three blocks. The left block, From Event Status, shows the possible initial statuses of an event. These statuses are Incomplete, Unprocessed, No Action, and Processed.

The center block, Action, represents actions that you complete in subledger applications. These actions result in events being created, processed, or deleted.

The right block, To Event Status, shows the possible final status based on the action in the central block. The status values include Incomplete, Unprocessed, No Action, and Processed Program.

All of the possible status changes are displayed in the diagram. Not all accounting event types can support all of these status changes. For example, some accounting events, once they have a status of Unprocessed, cannot be updated to an Incomplete status. Implementers are responsible for determining the supported status changes for an event.

There may also be conditions that determine whether the accounting event can move from one status to another. These conditions can vary by accounting event.

For example, a loan that has not been processed for accounting may be cancelled and will be set to a No Action event status. However, if unprocessed loan interest accrual events cannot be cancelled, the event status cannot be set to a No Action status.

## Event Date: Explained

Each accounting event must have one and only one event date. If there are multiple dates for a particular event type, then one accounting event must be created for each accounting date.

For example assume a loan is originated and accounted. The next day the loan interest is adjusted. Later, a loan payment is scheduled.

This creates three accounting events:



If the transaction has the potential to create multiple events, then both the event date and status of the previous event determines whether a new accounting event is created. Consider the following examples:

- A user cancels the loan before accounting for the loan origination event.
- If the event date for the cancellation is the same as that of the accounting event created for the loan, then the status of the loan origination accounting event is updated to No Action. No accounting event is created for the cancellation.
- An loan origination is canceled after it has been accounted.
- The accounting event created for the loan origination is not affected by the cancellation. A new event for the cancellation is created and requires processing to create a subledger journal entry.

## Accounting Event Order: Explained

All events have an associated accounting event type. For transactions that are document based, event types correspond to the different operations that can be performed on a document.

### Accounting Event Order

If there is an implied order in the accounting events for the different operations on a particular document, then implementers are responsible for enforcing that

order. The Oracle Fusion Accounting Hub is neither aware of, nor does it enforce any implied order.

### Example: Implied Order in Custom Loan Application

Assume the following three event types for a loan: Loan Originated, Loan Payment Scheduled, and Loan Interest Accrued.

It should not be possible to generate a subledger journal entry to account for an loan payment before the originated event is processed. Similarly, it should not be possible to adjust the loan after it has been canceled. Therefore there is an implied order to these event types.

### Correctly Enforcing Implied Order

Assume that the Loan application has four accounting events for a payment as described in the table below:

Event Number	Event Type	Event Date
1	Loan Originated	August 1, 2012
2	Loan Payment Scheduled	August 15, 2012
3	Loan Payment Adjusted	August 16, 2012
4	Loan Payment Canceled	August 28, 2012

In this case, the Loan application has correctly ordered the accounting events. For example, since event processing is by event number, the Accounting Hub does not attempt to account for the adjustment or cancellation of the payment before the Loan Payment Scheduled accounting event has been processed.

The Loan Payment Canceled event must receive an event number greater than the event number of the last Loan Payment Adjusted event type. Similarly, the Loan Payment Adjusted events receive event numbers greater than the Loan Payment Scheduled event but less than the Loan Payment Canceled event.

### No Implied Order Between Events of the Same Type

In the following example, assume that event number 3 has an earlier date than event number 2 as described in the table below. Note that event number 2 and event number 3 have the same type of Loan Payment Adjusted.

Event Number	Event Type	Event Date
1	Loan Payment Scheduled	August 1, 2012
2	Loan Payment Adjusted	August 16, 2012
3	Loan Payment Adjusted	August 7, 2012
4	Loan Payment Canceled	August 28, 2012

If the user submitted the Create Accounting process with a To Date of August 10th, event 1 and then event 3 are processed. Events 2 and 4 remain unprocessed. The latter two events are selected during a later submission of the Create Accounting process with a To Date greater than August 28th. Since there is no implied order between the two Loan Payment Adjusted accounting events, this is acceptable ordering.

If the user submitted the Create Accounting process with a To Date of the August 31st, all events are processed in event number order, with the August 7th adjustment being processed after the August 16th adjustment. Therefore, the order that the events are processed is affected by the To Date when submitting the Create Accounting process.

### **Non Enforcing the Implied Order, Incorrect Dating of Events**

The tables below describes this example:

Event Number	Event Type	Event Date
1	Loan Payment Scheduled	August 1, 2012
2	Loan Payment Adjusted	August 15, 2012
3	Loan Payment Adjusted	August 16, 2012
4	Loan Payment Canceled	July 20, 2012

In this example, the events are appropriately numbered. If the user specified a To Date equal to or later than August 16th, the events would still be processed in the correct order.

However, the events are not appropriately dated. If the user specified a To Date less than August 16th, the Create Accounting process would be unaware of the Loan Payment Adjusted event number 3, and the Loan Payment Canceled event on July 20th would be processed before the adjusted event.

In this example, the Loan application has incorrectly ordered the accounting events using the event dates. Therefore, it is possible to account for the cancellation of a loan payment before the Loan Payment accounting event has been processed.

## **Register Source Systems**

### **Register Source Systems: Critical Choices**

Subledger applications can support third party control account type and calculate reporting currency amounts.

#### **Calculate Reporting Currency Amount**

If the subledger application is configured to calculate reporting currency amount, there is no need to provide reporting currency information in the transaction objects.

#### **Additional Considerations**

The following are additional considerations when creating a subledger application:

1. Determine the subledgers requirement. For example, how many subledgers are to be created? This may depend on what security your company wants to have over its accounting rules.
  - Using the same subledger allows you to share subledger accounting rules, and lets you report across all data easily.

- Using separate subledgers provides more security across applications and less data in each process run providing better performance. Specific benefits are:
  - If you run two Create Accounting requests at the same time for different applications, they are much less likely to contend for database resources. The requests will perform better, as the indexes are tuned for running with different applications instead of running for different process categories within the same application.
  - It allows you to efficiently process different sets of data (different applications) at different times during the day instead of running it as one process.
- 2. Determine the transaction objects requirements. These requirements determine what source data is required to successfully create subledger journal entries from transactions that are captured in transaction objects and shared in reference objects.
- 3. Analyze accounting events to determine what events to capture for the subledger application.

Create programs to capture accounting events using APIs (application programming interfaces) that are provided as follows:

- Get Event Information APIs to get event information related to a document or a specific event.
  - Create Event APIs to create accounting events, individually or in bulk.
  - Update Event APIs to update events and keep them consistent with related transaction data.
  - Delete Event APIs to delete events.
4. Determine how often to capture accounting events, populate transaction objects, and run the Create Accounting process. This may depend on the immediacy and volumes of accounting requirements in your company.

## Accounting Event Model: Explained

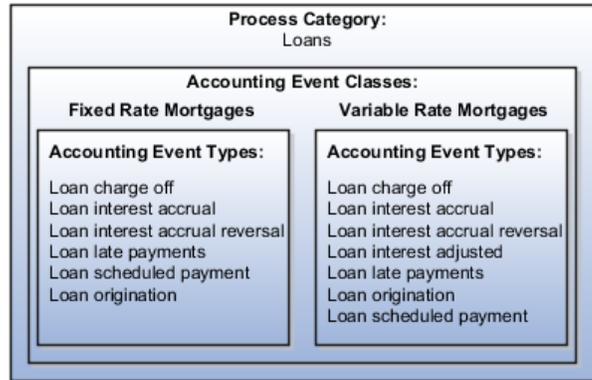
Accounting events represent transactions that may have financial significance, for example, issuing a loan and disposing of an asset. Financial accounting information can be recorded for these events and accounted by the Create Accounting process. When you define accounting events, determine from a business perspective which activities or transactions that occur in your source system may create a financial impact.

Events with significantly different fiscal or operational implications are classified into different accounting event types. Event types are categorized into accounting event classes. Accounting definitions in the Oracle Fusion Accounting Hub are based on event types. An event type must be unique within an application, process category, and event class.

Events are captured when transactions are committed in the subledgers, or they may be captured during end-of-day or period-end processing. For example, a

loan is originated, possibly adjusted, interest is accrued, and then the loan is paid or canceled. The accounting events representing these activities can create one or more subledger journal entries, and subsequently link the originating transaction to its corresponding journal entries.

The following is an example of an accounting event model for a loan application:



## Process Categories

A process category consists of specific event classes and the event types within those classes. To restrict the events selected for accounting, users can select a process category when they submit the Create Accounting process.

## Event Classes

You can assign a transaction view, system transaction identifiers, and optionally user transaction identifiers and processing predecessors for an event class in the Edit Event Class section. The transaction view should include all columns that have been mapped to system transaction identifiers for the accounting event class as well as the user transaction identifiers.

## System Transaction Identifiers

System transaction identifiers provide a link between an accounting event and its associated transaction or document. An identifier is the primary key of the underlying subledger transaction, usually the name of the surrogate key column on the transaction (header) associated with the accounting event. At least one system transaction identifier must be defined for the accounting event class.

When an accounting event is captured, system transaction identifiers, along with other required event data, are validated for completeness.

## User Transaction Identifiers

User transaction identifiers constitute the user-oriented key of the underlying subledger transaction, and are typically drawn from one or more database tables. These identifiers are primarily used in accounting events inquiry and on accounting event reports, to uniquely identify transactions. You can specify up to ten columns from the transaction views that are available for inquiry and reports.

The transaction data that identifies the transaction varies by accounting event class. Accounting event reports and inquiries display the transaction identifiers and their labels appropriate for the corresponding event class. The user transaction identifiers can be displayed for an event regardless of its status.

This includes the case where the accounting event has not been used to create subledger journal entries due to an error or the cases where it has not been processed. The user transaction identifier values are displayed at the time that the accounting event reports and inquiries are run. If a transaction identifier value has changed after the accounting event was captured, the accounting event reports and inquiries reflect the change.

### **Processing Predecessors**

The processing predecessor establishes an order in which the Create Accounting process processes events selected for accounting.

### **Event Types**

For accounting event types, specify whether their accounting events have accounting or tax impact. When the Create Accounting process is submitted, it only accounts business events that are enabled for accounting.

## **Transaction Objects: Points to Consider**

You may assign transaction and reference objects for each accounting event class in the subledger application. Sources are generated based on the transaction objects and are assigned to the corresponding accounting event classes.

Sources are used to create accounting rules. Subledgers pass information to the application by populating transaction object tables. The columns in these tables are named after the source codes. Transaction and reference objects hold transaction information that is useful when creating journal entry rules for accounting. The transaction and reference objects are defined for an accounting event class so that source assignments to accounting event class can be generated using these objects.

### **Transaction Objects**

Transaction objects refer to the tables or views from which the Create Accounting process takes the source values to create subledger journal entries. Source values, along with accounting event identifiers, are stored in the transaction objects. The Create Accounting process uses this information to create subledger journal entries.

You have several options. You can:

- Create new tables as the transaction objects and create a program to populate them.
- Use views of your transaction data as the transaction objects.
- Use your transaction data tables as the transaction objects.

The transaction objects and or views must be accessible to the Create Accounting process. Typically, an ETL (extract, transformation, and load) program is used to take values from the source system and load them into the database used by the Create Accounting process. The ETL process is done outside of the Create Accounting process.

The following are transaction object types:

- Header transaction objects

- Implementers need to provide at least one header transaction object for each accounting event class. Header transaction objects store one row with untranslated header source values for each accounting event. The primary key of a header transaction object is the event identifier.

Transaction details that are not translated, and whose values do not vary by transaction line or distribution, should normally be stored in header transaction objects. Examples of sources normally stored in header transaction objects include the Loan Number for a loan or the Contract Number for a contract.

- Line transaction objects
  - Line transaction objects are relevant when there are details for the accounting event that vary based upon transaction attributes. For example, a mortgage transaction for loan origination may have multiple amounts, each related to different components of the loan. There may be a loan origination amount, closing cost amounts, and escrow amounts. Each of these amounts could be captured as separate lines, along with an indication of the amount type

Line transaction objects store untranslated line level source values. There should be one row per distribution, identified by a unique line number. The primary key of a line transaction object is the event identifier and transaction object line number. Transaction details that are not translated and whose values vary by transaction line or distribution are normally stored in line transaction objects columns. Examples include the Loan Number for a loan payment.

- Multi-Language Support (MLS) transaction objects
  - MLS transaction objects are relevant to applications that support the MLS feature. MLS transaction objects store translated source values. There should be one row per accounting event and language. The primary key of a header MLS transaction object is the event identifier and language. The primary key of a line MLS transaction object is the event identifier, transaction object line number, and language.

Transaction details that are translated, and whose values do not vary by transaction line or distribution, are normally stored in header MLS transaction object columns. Examples include Loan Terms for a commercial loan. Implementers can avoid having to store source values in header MLS transaction objects by using value sets and lookup types.

Transaction details that are translated, and whose values vary by transaction line or distribution, should normally be stored in the transaction object in columns defined in a line MLS transaction object.

## Reference Objects

Reference objects are useful for storing information that is used for creating subledger journal entries. This information may not be directly from the source system or may be used for many entries, thus making it redundant. Use reference objects to share sources information across applications.

For example, store customer attributes, such as the customer class or credit rating in a reference object, and then, use it to account for different journal entries in a loan cycle, such as loan origination or interest accrual. Store information, such as bond ratings and terms, and use it to account for entries across the life of bonds, such as interest accruals or bond retirement.

Reference objects can either have a direct relationship to transaction objects (primary reference object), or be related to other reference objects (secondary).

## Transaction Objects Guidelines: Points to Consider

This section provides guidelines to help determine what transaction data should be stored in the transaction objects.

### Accounting Reversals: Points to Consider

To invoke an accounting reversal for a transaction or transaction distribution, the transaction objects should include the appropriate header or line level accounting reversal options.

Accounting reversal enables users to reverse the accounting impact of a previously accounted transaction distribution or all existing accounting for a transaction.

The Oracle Fusion Accounting Hub uses the following accounting reversal specific terminology.

### Reversed (Original) Distribution

Refers to a transaction distribution that although successfully accounted, is either incorrect or canceled. The transaction distribution is therefore reversed.

### Reversal Distribution

Refers to a transaction distribution which reverses the effect of the original distribution on transaction balances. Typically, reversal distributions are identical to the reversed distributions in all respects except for entered (ledger) amounts that reverse the sign of the original.

### Replacement Distribution

Refers to a transaction distribution which replaces the reversed distribution with the correct value.

The table below illustrates the distributions described above. Distribution examples:

Invoice Distribution Line Number	Invoice Line Type	Accounting Date	Amount	Description
1	Item	10-Jan-2012	1000	Reversed
2	Item	12-Jan-2012	-1000	Reversal (of line 1)
3	Item	12-Jan-2012	2000	Replacement (of line 1)

Transaction Total			2000	
-------------------	--	--	------	--

Note that the original accounting impact of the reversed distributions is undone, even if the subledger journal setups or accounting configurations have changed since the original subledger journal entry was generated.

## Managing Accounting Sources: Critical Choices

Sources are a key component for setting up accounting rules. Sources represent transaction and reference information from source systems. Contextual and reference data of transactions that are set up as sources can be used in accounting rules definitions.

When determining what sources should be available, it is helpful to begin the analysis by considering which information from your systems is accounting in nature. Examples of sources that are accounting in nature include general ledger accounts that are entered on transactions, the currency of a transaction, and transaction amounts. Sources that are not always required for accounting rules include items that are related to the transaction for other purposes than accounting. Examples of information that may not be specifically for accounting, but which may be useful for creating subledger journal entries, are transaction identification number (loan number, customer number, or billing account number), counter party information, and transaction dates.

Provide a rich library of sources from your source systems for maximum flexibility when creating definitions for subledger journal entries.

Sources are assigned to accounting event classes by submitting the Create and Assign Sources process.

There is a distinct difference between sources and source values. Sources represent the transaction attributes used to create accounting rules. Source values are used by the Create Accounting process to create subledger journal entries based upon source assignments to accounting rules.

### Sources

Sources must be created prior to creating accounting rules. This is a predefined step which must be undertaken before the application can be used to create subledger journal entries.

To set up appropriate subledger journal entry rule sets, users and those implementing need to understand the origins, meaning, and context of sources. Use business oriented names for sources to allow accountants and analysts to effectively create accounting rules.

- Enables users to easily identify a source.
- Ensures consistency in nomenclature.

### Source Values

Source values are stored in transaction objects. They are the actual transaction attribute values from the source system and are used in creation of the journal entries.

## Accounting Attribute Assignments: Points to Consider

The Create Accounting process uses the values of sources assigned to accounting attributes plus accounting rules to create subledger journal entries. Almost all accounting attributes have sources assigned at the accounting event class level. Depending on the accounting attribute, the accounting attribute assignment defaulted from the accounting event class can be overridden on journal line rules or subledger journal entry rule sets.

Once sources are assigned to accounting event classes, they are eligible for assignment to accounting attributes for the same accounting event classes.

The Create Accounting process uses these assignments to copy values from transaction objects to subledger journal entries. For example, you may map the invoice entered currency to the subledger journal entry entered currency.

Each accounting attribute is associated with a level:

1. Header: To be used when creating subledger journal entry headers.
2. Line: To be used when creating subledger journal entry lines.

The types of accounting attributes values are as follows:

### Values that are Subject to Special Processing

You may have values that are subject to special processing or values that are stored in named columns in journal entry headers and lines.

Examples of accounting attributes are Entered Currency Code and Entered Amount.

### Values that Control the Behavior of the Create Accounting Process

You may have values that control the behavior of the Create Accounting process when processing a specific accounting event or transaction object line.

An example of accounting attributes of this type is Accounting Reversal Indicator.

### Minimum Required Accounting Attribute Assignments

In order to create a valid journal entry you must, at a minimum, set up the following accounting attribute assignments.

- Accounting Date
- Distribution Type
- Entered Amount
- Entered Currency Code
- First Distribution Identifier

The details and descriptions of these attributes are included in the Accounting Attributes section.

## Accounting Attributes

Accounting attribute groups are represented in the tables below:

### Accounted Amount Overwrite

- The accounted amount overwrite accounting attribute indicates whether the accounted amount calculated by the Create Accounting process should be overwritten by the value of the accounted amount accounting attribute. If the source value mapped to Accounted Amount Overwrite is 'Y', then an accounted amount must be provided.

Accounting Attributes	Data Type	Journal Entry Level	Assignment to Rules	Assignment Required?	Validation Rules
Accounted Amount Overwrite Indicator	Alphanumeric	Line	Event Class and Journal Line Rule	No	Y - Overwrite accounted amount  N - Not overwrite accounted amount

### Accounting Date

- The accounting date attribute is relevant to all applications. The Create Accounting process uses it to derive the accounting date of journal entries. Typically, the event date system source is assigned to the accounting date attribute.
- The Accrual Reversal GL Date accounting attribute is relevant to applications using the accrual reversal feature. Users can assign system and standard date sources to the Accrual Reversal GL Date in the Accounting Attribute Assignments page. When the Accrual Reversal GL Date accounting attribute returns a value, the Create Accounting process generates an entry that reverses the accrual entry.

Accounting Attributes	Data Type	Journal Entry Level	Assignment to Rules	Assignment Required?	Validation Rules
Accounting Date	Date	Header	Event Class and Journal Entry Rule Set	Yes	Should be in open general ledger period
Accrual Reversal GL Date	Date	Header	Event Class and Journal Entry Rule Set	No	Should be later than the accounting date

### Accounting Reversal

- Accounting reversal accounting attributes are relevant to applications that wish to take advantage of the accounting reversal feature. The Create Accounting process uses them to identify transaction (distributions) whose accounting impact should be reversed. For the Create Accounting process to successfully create a line accounting reversal, the accounting reversal indicator, distribution type, and first distribution identifier should always be assigned to sources. The definition of the accounting reversal distribution type and distribution identifiers mirrors the definition of the distribution identifiers.

Accounting Attributes	Data Type	Journal Entry Level	Assignment to Rules	Assignment Required?	Validation Rules
Accounting Reversal Distribution Type	Alphanumeric	Line	Event Class	Yes, if another accounting reversal accounting attribute is assigned.	
Accounting Reversal First Distribution Identifier	Alphanumeric	Line	Event Class	Yes, if another accounting reversal accounting attribute is assigned.	
Accounting Reversal Second Distribution Identifier	Alphanumeric	Line	Event Class	No	
Accounting Reversal Third Distribution Identifier	Alphanumeric	Line	Event Class	No	
Accounting Reversal Fourth Distribution Identifier	Alphanumeric	Line	Event Class	No	
Accounting Reversal Fifth Distribution Identifier	Alphanumeric	Line	Event Class	No	
Accounting Reversal Indicator	Alphanumeric	Line	Event Class	Yes, if another accounting reversal accounting attribute is assigned.	Y - Reverse without creating a replacement line B - Reverse and create a new line as replacement N or Null - Not a reversal
Transaction Accounting Reversal Indicator	Alphanumeric	Header	Event Class	No	Y - Reversal transaction object header N or null - Standard transaction object header

#### Business Flow

- The business flow accounting attributes are referred to as 'applied to' accounting attributes. If a transaction is applied to a prior transaction in the business flow, the transaction object must populate sources assigned

to 'applied to' accounting attributes with sufficient information to allow the Create Accounting process to uniquely identify a transaction object line for a prior event in the business flow. When deriving accounting data from a previous event in the business flow, the Create Accounting process searches for a journal entry line for the prior event using a combination of the 'applied to' accounting attributes and the business flow class of both journal entries.

The Applied to Amount accounting attribute is used to calculate the accounted amount and gain or loss in cross-currency applications when business flows are implemented. This attribute value is used to calculate the accounted amount when a source is mapped to the Applied to Amount attribute on a journal line type and the entered currency is different than the original currency entered.

---

**Note**

When enabling business flow to link journal lines in the Journal Line Rule page, certain accounting attribute values are unavailable for source assignment in the Accounting Attributes Assignments window of the same page because they will be copied from the related prior journal entry.

---

Accounting Attributes	Data Type	Journal Entry Level	Assignment to Rules	Assignment Required?	Validation Rules
Applied to Amount	Number	Line	Event Class and Journal Line Rule	No	
Applied to First System Transaction Identifier	Alphanumeric	Line	Event Class and Journal Line Rule	Yes, if another accounting attribute in the same group has assignment.	
Applied to Second System Transaction Identifier	Alphanumeric	Line	Event Class and Journal Line Rule	No	
Applied to Third System Transaction Identifier	Alphanumeric	Line	Event Class and Journal Line Rule	No	
Applied to Fourth System Transaction Identifier	Alphanumeric	Line	Event Class and Journal Line Rule	No	
Applied to Distribution Type	Alphanumeric	Line	Event Class and Journal Line Rule	Yes, if another accounting attribute in the same group has assignment.	
Applied to First Distribution Identifier	Alphanumeric	Line	Event Class and Journal Line Rule	Yes, if another accounting attribute in the same group has assignment.	

Applied to Second Distribution Identifier	Alphanumeric	Line	Event Class and Journal Line Rule	No	
Applied to Third Distribution Identifier	Alphanumeric	Line	Event Class and Journal Line Rule	No	
Applied to Fourth Distribution Identifier	Alphanumeric	Line	Event Class and Journal Line Rule	No	
Applied to Fifth Distribution Identifier	Alphanumeric	Line	Event Class and Journal Line Rule	No	
Applied to Application ID	Number	Line	Event Class and Journal Line Rule	Yes, if another accounting attribute in the same group has assignment.	Must be a valid application ID
Applied to Entity Code	Alphanumeric	Line	Event Class and Journal Line Rule	Yes, if another accounting attribute in the same group has assignment.	Must be a valid Entity for the application selected in Applied to Application ID

#### Distribution Identifier

- Distribution identifiers accounting attributes are relevant to all applications. The distribution identifier information links subledger transaction distributions to their corresponding journal entry lines. In addition, many of the Oracle Fusion Subledger Accounting features, including accounting reversals, rely on the correct definition and storing of distribution identifiers in the line transaction objects. The distribution type and first distribution identifiers are always assigned to sources. If a transaction distribution is identified by a composite primary key, additional distribution identifiers are assigned to standard sources, as appropriate. Values for the distribution type and distribution identifiers are always stored in accounting transaction objects. The combinations of the values of the system transaction identifiers with the values of the distribution identifiers uniquely identify a subledger transaction distribution line.

Accounting Attributes	Data Type	Journal Entry Level	Assignment to Rules	Assignment Required?	Validation Rules
Distribution Type	Alphanumeric	Line	Event Class	Yes	
First Distribution Identifier	Alphanumeric	Line	Event Class	Yes	
Second Distribution Identifier	Alphanumeric	Line	Event Class	No	

Third Distribution Identifier	Alphanumeric	Line	Event Class	No	
Fourth Distribution Identifier	Alphanumeric	Line	Event Class	No	
Fifth Distribution Identifier	Alphanumeric	Line	Event Class	No	

#### Document Sequence

- The document sequence accounting attributes are relevant to applications that use the document sequencing feature to assign sequence numbers to subledger transactions. The Create Accounting process uses them to provide a user link between subledger transactions and their corresponding subledger journal entries. Assign all document sequence accounting attributes to sources or do not assign any. In addition, the Document Sequence Category Code is made available as an Accounting Sequence Numbering control attribute.

Accounting Attributes	Data Type	Journal Entry Level	Assignment to Rules	Assignment Required?	Validation Rules
Subledger Document Sequence Category	Alphanumeric	Header	Event Class	Yes, if another accounting attribute in the same group has assignment.	
Subledger Document Sequence Identifier	Number	Header	Event Class	Yes, if another accounting attribute in the same group has assignment.	
Subledger Document Sequence Value	Number	Header	Event Class	Yes, if another accounting attribute in the same group has assignment.	

#### Entered Currency

- Entered currency accounting attributes are relevant to all applications. The Create Accounting process uses them to populate the journal entry line entered currency code and amounts. The entered currency accounting attributes must always be assigned to sources. The sources assigned to the entered currency accounting attributes must always contain a value. For event classes that support cross currency transactions and therefore, more than one entered currency and entered currency amount, multiple event class accounting attribute assignments are created.

Accounting Attributes	Data Type	Journal Entry Level	Assignment to Rules	Assignment Required?	Validation Rules
Entered Currency Code	Alphanumeric	Line	Event Class and Journal Line Rule	Yes	A valid currency code

Entered Amount	Number	Line	Event Class and Journal Line Rule	Yes	
----------------	--------	------	-----------------------------------	-----	--

### Ledger Currency

- Ledger currency accounting attributes are relevant to all applications that use the Create Accounting process. The Create Accounting process uses them to populate journal entry accounted amounts. If a transaction's entered currency is different from the ledger currency, the Create Accounting process copies the conversion date, conversion rate, and conversion rate type to the corresponding journal entry lines. If the entered currency is the same as the ledger currency, the Create Accounting process ignores the conversion type and conversion rate. For event classes that support foreign currency transactions and therefore more than one exchange rate and reporting currency amount, multiple event class accounting attribute assignments are created.

Accounting Attributes	Data Type	Journal Entry Level	Assignment to Rules	Assignment Required?	Validation Rules
Accounted Amount	Number	Line	Event Class and Journal Line Rule	No	
Conversion Date	Date	Line	Event Class and Journal Line Rule	No	
Conversion Rate	Number	Line	Event Class and Journal Line Rule	No	
Conversion Rate Type	Alphanumeric	Line	Event Class and Journal Line Rule	No	A valid general ledger conversion rate type or User

### Tax

- The tax accounting attributes are relevant to applications that uptake the tax initiative. The tax team uses the tax accounting attributes to link subledger transaction tax distributions to their corresponding journal entry lines. Oracle Fusion Tax specifies which tax reference values are mandatory in transaction objects and are assigned to standard sources.

Accounting Attributes	Data Type	Journal Entry Level	Assignment to Rules	Assignment Required?	Validation Rules
Detail Tax Distribution Reference	Number	Line	Event Class	No	
Detail Tax Line Reference	Number	Line	Event Class	No	
Summary Tax Line Reference	Number	Line	Event Class	No	

### Third Party

- Third party accounting attributes are relevant to subledger applications that use third party control accounts. The third party accounting attributes link suppliers and customers to their corresponding subledger journal entry lines in the supplier and customer subledgers. For all subledger transactions that represent financial transactions with third parties, all third party accounting attributes have sources assigned. If a transaction line is associated with a customer or supplier, the transaction objects need to include values for all sources mapped to third party accounting attributes for the event class.

Accounting Attributes	Data Type	Journal Entry Level	Assignment to Rules	Assignment Required?	Validation Rules
Party Identifier	Number	Line	Event Class and Journal Line Rule	Yes, if another accounting attribute in the same group has assignment.	If party type C - Should be a valid customer account  If party type is S - Should be a valid supplier identifier
Party Site Identifier	Number	Line	Event Class and Journal Line Rule	Yes, if another accounting attribute in the same group has assignment.	If party type C - Should be a valid customer account  If party type is S - Should be a valid supplier identifier
Party Type	Alphanumeric	Line	Event Class	Yes, if another accounting attribute in the same group has assignment.	C for Customer  S for Supplier

### Exchange Gain Account, Exchange Loss Account

- The Create Accounting process determines whether there is an exchange gain or loss and derives the account combination based on whether the journal line rule is defined. If the gain or loss journal line rule is defined, the account rule assigned to the journal line rule is used to determine the gain or loss account to use. If the gain or loss journal line rule is not defined, the gain or loss account assigned to the Exchange Gain Account and Exchange Loss Account accounting attributes is used.

Accounting Attributes	Data Type	Journal Entry Level	Assignment to Rules	Assignment Required?	Validation Rules
Exchange Gain Account	Number	Header	Event Class	No	
Exchange Loss Account	Number	Header	Event Class	No	

### Gain or Loss Reference

- The Gain or Loss Reference accounting attribute groups entry lines together when calculating exchange gain or loss. The accounted debit and accounted credit amounts for lines with the same gain or loss reference are combined. The total of accounted debit and total of accounted credit are compared to calculate the exchange gain or loss.

Accounting Attributes	Data Type	Journal Entry Level	Assignment to Rules	Assignment Required?	Validation Rules
Gain or Loss Reference	Alphanumeric	Line	Event Class	No	

### Transfer to GL Indicator

- The Transfer to GL accounting attribute is relevant to applications which create subledger journal entries that will never be transferred to the general ledger. The Transfer to GL process uses this accounting attribute to determine whether to transfer subledger journal entries to the general ledger.

If the Transfer to GL accounting attribute is not assigned to a source, the Transfer to GL process transfers journal entries for the event class to the General Ledger.

If the Transfer to GL accounting attribute is assigned to a source and the source is not populated, the Transfer to GL process transfers journal entries for the event class to the General Ledger.

Accounting Attributes	Data Type	Journal Entry Level	Assignment to Rules	Assignment Required?	Validation Rules
Transfer to GL Indicator	Alphanumeric	Header	Event Class	No	Should be Y or N

## Implement Accounting Event Interfaces

### Accounting Event Interfaces: Overview

The Oracle Fusion Accounting Hub provides a set of common APIs to capture accounting events. All event operations must be performed through these APIs. By ensuring that event operations are executed through generic APIs, the architecture meets the needs of implementers:

- Insulates implementers from changes in the implementation of an API

The presence of the API reduces dependencies between the Accounting Hub and your source systems.

Implementers do not have to know the underlying Accounting Hub data model to capture accounting events. In addition, any implementation changes made by the Accounting Hub have minimum or no impact on implementers.

The Accounting Hub relies on accounting events to indicate that there are activities from source systems that require accounting.

The Create Accounting process selects accounting events based on criteria specified by users. The Create Accounting process does not check for any functional dependencies between transactions or event types.

---

**Note**

For each eligible event, the Create Accounting process retrieves source values from the transaction objects. Subledger journal entries are created using both event and source information.

---

## Procedures to Create Event Capture Routines: Explained

Implementers must undertake the following steps to create event capture routines:

- Perform accounting event setups
- Write product specific wrapper routines
- Integrate event APIs with source systems

### Perform Event Setups

As a prerequisite, you must register the application.

An additional prerequisite step is to define and register event process categories, accounting event classes, and event types before events can be captured. The event APIs use event information to perform the necessary validations when creating events.

### Write Product Specific Cover Routines

In order to reduce dependencies and facilitate maintenance, it is recommended that you write a wrapper routine on top of the Oracle Fusion Accounting Hub APIs. Wrapper routines can encapsulate source system specific logic to perform necessary validations before creating new events. Map source system specific parameters to the API parameters

For example, if you were implementing the Accounting Hub to capture information for a loans application, you could use a package called `LOAN_XLA_EVENTS_PKG`, which contains all the APIs to implement accounting events in Loans. The code for this package is shown below.

Assume that this example loans application has two event classes Create Loan and Create Payment. To handle accounting events for loan transactions, you could create a procedure `create_loan_event()` with `p_loan_id` as an input parameter, instead of using a generic parameters like `source_id_int_1`. The procedure `create_loan_event()` calls the appropriate Accounting Hub API to create an event.

Similarly, create `create_payment_event()` to handle accounting events for the event class Create Payment.

```
LOAN_XLA_EVENTS_PKG
```

```

-- Procedure to create events for creating a loan
PROCEDURE create_loan_event
(p_loan_id
,p_event_type
,p_event_date
,p_event_status) IS
l_loan_source_info XLA_EVENTS_PUB_PKG.t_event_source_info;
l_security_context XLA_EVENTS_PUB_PKG.t_security; BEGIN
BEGIN
-- Perform product specific checks
...
-- Map generic event API parameter to the product specific columns
l_loan_source_info.application_id = loan_application_id;
l_loan_source_info.legal_entity_id = l_legal_entity_id;
l_loan_source_info.source_id_int_1 = p_loan_id;
-- Call XLA API
XLA_EVENTS_PUB_PKG.create_event
(p_event_source_info => l_loan_source_info
,p_event_type_code => p_event_type
,p_event_date => p_event_date
,p_event_status_code => p_event_status
,p_event_number => NULL
,p_reference_info => NULL
,p_valuation_method => NULL
,p_security_context => l_security_context);
...
...
EXCEPTIONS
....
END;

```

## Integrate Event APIs with Source Systems

It is suggested that implementers create an Enterprise Scheduler Service (ESS) process for the event capture wrapper routines. Using an ESS process instead of directly running the routines in the database will provide for the following:

- Enable implementation of security where some users can be given access to create events for specific source systems but not all.
- Go through the proper channels of executing code on Fusion schema. The ESS job will not require the user to have write access to Fusion schema.

## Overview of Event APIs

### Getting Event Information: Overview

You may need to perform certain checks with respect to events. For example, before creating a new accounting event, it is necessary to check whether there is an existing unprocessed event for the same transaction with the same accounting event type and event date.

You may also want to know the status of a particular event or query events already created for the transaction.

To perform these checks, obtain event information for a transaction by doing the following:

1. Determine the system transaction identifiers and accounting event class for the transaction.

---

#### Note

System transaction identifiers identify the transaction on which events are based. The Oracle Fusion Accounting Hub uses these identifiers to search the events table and identify all the events that are related to a transaction.

- 
2. Call the function `Get_Array_Event_Info()` with the appropriate transaction parameters. The function returns an array of all events for a transaction.

To obtain all the events created for a particular event type within a transaction, do the following:

1. Determine the system transaction identifiers, event class, and event type of the transaction.
2. Call the function `Get_Array_Event_Info()` with the appropriate transaction and event type input parameters. The function returns an array of all accounting events for that transaction and event type. Optionally pass the event class, event date, and event status to further restrict the rows returned.

To get information about a specific event, do the following:

1. Determine the `event_id`.
2. Call the function `Get_Event_Info()` with the `event_id` parameter. This function returns a PL/SQL record containing all information pertaining to that particular event.

## Creating Events: Explained

This section describes the following guidelines on creating events using the create event APIs:

- Creating a single event
- Creating events in bulk

### Creating a Single Event

To create a new event:

1. Determine the accounting event type, event date, and event status for the new event.
2. Call the Oracle Fusion Accounting Hub function `Create_Event()` with the appropriate input parameters.

The `Create_Event()` API creates a single event at a time. The function returns the `event_id` of the created event.

### Creating Events in Bulk

Create events in bulk using the API `Create_Bulk_Events()`.

---

#### Note

Do not use this API for existing transactions that already have events associated with them. For performance reasons, bulk event APIs do not perform checks as

to whether events for the transaction already exist. Therefore, use this API only to create events for new transactions that do not have any prior events created.

---

## Updating Events: Overview

Update the event to keep the transaction data and related events synchronized.

You can update an event as long as it is not processed. Once an event is accounted in Final status, you cannot update the event or the data associated with it.

Use the following APIs to update your events:

- Update Event  
This is an overloaded API used to update the status, type, date, number, and reference information of a single event.
- Update Event Status  
This API updates multiple events to a given status.

## Deleting Events: Explained

To delete all Unprocessed events associated with a transaction:

1. Determine the transaction source information.
2. Call the Oracle Fusion Accounting Hub function `Delete_Events()` with the transaction source information.
3. Optionally, specify the accounting event type, event status, or event date, to restrict the events deleted.

If a transaction is deleted, the `Delete_Entity()` API must be called to delete the row in the `XLA_TRANSACTIONS_ENTITY` table.

## Updating Transaction Numbers: Overview

In the case where the transaction number of the transaction has been changed, this API updates the transaction number on the events in the `XLA_TRANSACTION_ENTITIES` table so that they are consistent with the transaction number on the transaction.

The API checks the source information for a valid application, legal entity, event process category, and source identifiers. However, no validation is performed on the transaction number.

## Details of Event API

### Event API Details: Overview

This section describes the event APIs accessible by implementers to perform event operations. The APIs described are generic and available to all applications.

Event APIs have the following characteristics:

- Event APIs do not issue any commits.
- If an API fails, any changes made are rolled back and a standard exception is raised.
- All parameters are read only (IN parameters); there are no OUT parameters.
- The event date is always truncated.

The Oracle Fusion Accounting Hub does not store the timestamp for an event date.

- All functions return a single value or record.

The exceptions to this rule are the functions prefixed with a `Get_Array` string. For example, the function `Get_Array_Event_Info()` returns an array.

- Input parameters must be passed in a nonpositional notation, named notation.
- All the APIs called in query mode locks the event record in NOWAIT mode.

Event APIs have the following types of input parameters:

- System transaction identifiers

These parameters capture information such as `loan_id`. This information is stored with each event to later identify the source transaction for the event. You need to pass the source identifiers when creating an event.

- Transaction security identifiers

Accounting events are subject to the security of the corresponding transaction. Every accounting event is stamped with its related transaction's security context.

Transaction security parameters capture application-specific transaction security information, such as business unit or ledger.

- Transaction reference information

The reference parameters enable you to capture any miscellaneous reference or contextual information for an event. This information is optional and no validations are performed against any reference parameters.

The `XLA_EVENT_PUB_PKG` package contains the following items:

- PL/SQL record and table structures for common parameters
- CONSTANTS for event statuses

Use these constants and structures when passing and reading values to and from the APIs.

## Getting Event Information APIs: Examples

This section provides details on the APIs that obtain event information.

## 1. XLA\_EVENTS\_PUB\_PKG.GET\_EVENT\_INFO()

This API returns information for a particular event. The event is identified by specifying the transaction and event identifier. The API locks the specified event before returning any event information.

The API checks all source information for valid application, legal entity, event entity, and source identifiers (IDs). It ensures that the required parameters are not passed as null and that the event ID belongs to the same transaction, as the other transaction information being passed.

The API returns a PL/SQL record containing event information.

```
FUNCTION XLA_EVENTS_PUB_PKG.get_event_info
(p_event_source_info IN xla_events_pub_pkg.t_event_source_info
,p_event_id IN NUMBER
,p_valuation_method IN VARCHAR2
,p_security_context IN xla_events_pub_pkg.t_security)
RETURN xla_events_pub_pkg.t_event_info;
```

## 2. XLA\_EVENTS\_PUB\_PKG.GET\_ARRAY\_EVENT\_INFO()

This routine returns information for one or more events within a transaction. The calling program specifies the transaction and event selection criteria. Return information contains data on all events that belong to the specified transaction and fall under the given criteria. The API locks the specified events before returning the event information.

The API checks all source information for valid application, legal entity, event process category, and source IDs. It ensures that the required parameters are not passed as null and also validates the accounting event class, event type, and event status. Note that the API truncates the event date.

The API returns an array of event information.

```
FUNCTION XLA_EVENTS_PUB_PKG.get_array_event_info
(p_event_source_info IN xla_events_pub_pkg.t_event_source_info
,p_event_class_code IN VARCHAR2 DEFAULT NULL
,p_event_type_code IN VARCHAR2 DEFAULT NULL
,p_event_date IN DATE DEFAULT NULL
,p_event_status_code IN VARCHAR2 DEFAULT NULL
,p_valuation_method IN VARCHAR2
,p_security_context IN xla_events_pub_pkg.t_security)
RETURN xla_events_pub_pkg.t_array_event_info;
```

## 3. XLA\_EVENTS\_PUB\_PKG.GET\_EVENT\_STATUS()

This API returns the event status for a specified event. The calling program needs to specify the transaction and event identifier. The API locks the specified event record before returning the status.

The API checks all source information for valid application, legal entity, event process category, and source IDs. It ensures that the required parameters are not null and the event belongs to the same transaction as the other transaction information being passed.

This API returns an event status. The Oracle Fusion Accounting Hub has defined all event statuses as Constants.

```
FUNCTION XLA_EVENTS_PUB_PKG.get_event_status
(p_event_source_info IN xla_events_pub_pkg.t_event_source_info
```

```
,p_event_id IN NUMBER
,p_valuation_method IN VARCHAR2
,p_security_context IN xla_events_pub_pkg.t_security)
RETURN VARCHAR2;
```

#### 4. XLA\_EVENTS\_PUB\_PKG.EVENT\_EXISTS()

This API checks whether an event exists for the specified criteria. It returns True if it finds at least one event matching the criteria; otherwise, it returns False. The API locks the event rows before returning a value.

The API checks all source information for valid application, legal entity, event process category, and source IDs. It ensures that the required parameters are not null and also validates the event class, event type, and event status. The API truncates the event date.

The API returns True if an event is found for the specified criteria; otherwise, it returns False.

```
FUNCTION XLA_EVENTS_PUB_PKG.event_exists
(p_event_source_info IN xla_events_pub_pkg.t_event_source_info
,p_event_class_code IN VARCHAR2 DEFAULT NULL
,p_event_type_code IN VARCHAR2 DEFAULT NULL
,p_event_date IN DATE DEFAULT NULL
,p_event_status_code IN VARCHAR2 DEFAULT NULL
,p_event_number IN NUMBER DEFAULT NULL
,p_valuation_method IN VARCHAR2
,p_security_context IN xla_events_pub_pkg.t_security)
RETURN BOOLEAN;
```

## Creating Event APIs: Examples

This section provides details on the API that create events.

#### 1. XLA\_EVENTS\_PUB\_PKG.CREATE\_EVENT()

This API creates a new event.

The API checks all source information for valid application, legal entity, event process category, and source IDs. It ensures that the required parameters are not null and also validates the accounting event type and event status.

No validations are performed against the reference columns and event number. However, if no event number is passed, the routine populates the next highest event number for that transaction. The event date is truncated.

If an event is created successfully, then the function returns its event ID.

```
FUNCTION XLA_EVENTS_PUB_PKG.create_event
(p_source_event_info IN xla_events_pub_pkg.t_event_source_info
,p_event_type_code IN VARCHAR2
,p_event_date IN DATE
,p_event_status_code IN VARCHAR2
,p_event_number IN NUMBER DEFAULT NULL
,p_reference_info IN xla_events_pub_pkg.t_event_reference_info
DEFAULT NULL
,p_valuation_method IN VARCHAR2
,p_transaction_date IN DATE DEFAULT NULL
,p_security_context IN xla_events_pub_pkg.t_security)
RETURN NUMBER;
```

```
FUNCTION XLA_EVENTS_PUB_PKG.create_event
```

```

(p_event_source_info IN xla_events_pub_pkg.t_event_source_info
,p_event_type_code IN VARCHAR2
,p_event_date IN DATE
,p_event_status_code IN VARCHAR2
,p_event_number IN NUMBER DEFAULT NULL
,p_transaction_date IN DATE DEFAULT NULL
,p_reference_info IN xla_events_pub_pkg.t_event_reference_info
DEFAULT NULL
,p_valuation_method IN VARCHAR2
,p_security_context IN xla_events_pub_pkg.t_security)
RETURN NUMBER;

```

## 2. XLA\_EVENTS\_PUB\_PKG.CREATE\_BULK\_EVENTS()

This API creates multiple events for multiple transactions.

---

### Note

Do not use this API for existing transactions that already have events associated with them. For performance reasons, bulk event APIs do not perform checks as to whether events for the transaction already exist. Therefore, use this API only to create events for new transactions that do not have any prior events created.

---

Information required for each event is inserted into the XLA\_EVENTS\_INT\_GT table as described below, before the API is called.

The API checks all source information for valid application, legal entity, event entity, event number, and source IDs. It ensures that the required parameters are not null and also validates the event type and event status.

No validations are performed against the reference columns and event number.

Column Name	Data Type	Size	Required
ENTITY_CODE	VARCHAR2	30	Yes
APPLICATION_ID	NUMBER	15	Yes
LEDGER_ID	NUMBER	15	Yes
EVENT_STATUS_CODE	VARCHAR2	30	Yes
EVENT_TYPE_CODE	VARCHAR2	30	Yes
EVENT_DATE	DATE	-	Yes
TRANSACTION_DATE	DATE	-	No
VALUATION_METHOD	VARCHAR2	30	No
TRANSACTION_NUMBER	VARCHAR2	240	No
BUDGETARY_CONTROL_FLAG	VARCHAR2	1	No
SOURCE_ID_INT_1	NUMBER	15	No
SOURCE_ID_INT_2	NUMBER	15	No
SOURCE_ID_INT_3	NUMBER	15	No
SOURCE_ID_INT_4	NUMBER	15	No
SOURCE_ID_CHAR_1	VARCHAR2	30	No
SOURCE_ID_CHAR_2	VARCHAR2	30	No
SOURCE_ID_CHAR_3	VARCHAR2	30	No

SOURCE_ID_CHAR_4	VARCHAR2	30	No
SECURITY_ID_INT_1	NUMBER	15	No
SECURITY_ID_INT_2	NUMBER	15	No
SECURITY_ID_INT_3	NUMBER	15	No
SECURITY_ID_CHAR_1	VARCHAR2	30	No
SECURITY_ID_CHAR_2	VARCHAR2	30	No
SECURITY_ID_CHAR_3	VARCHAR2	30	No
REFERENCE_NUM_1/ REFERENCE_NUM_4	NUMBER	15	No
REFERENCE_CHAR_1/ REFERENCE_CHAR_4	VARCHAR2	30	No
REFERENCE_DATE_1/ REFERENCE_DATE_4	DATE	-	No

```

Procedure XLA_EVENTS_PUB_PKG.create_bulk_events
(p_source_application_id IN NUMBER DEFAULT NULL
,p_application_id IN NUMBER
,p_legal_entity_id IN NUMBER DEFAULT NULL
,p_ledger_id IN NUMBER
,p_entity_type_code IN VARCHAR2);

```

### PL/SQL Requirements

When calling the Event API from a PL/SQL session, you must first initialize the session.

1.

Run the below query and note the **role\_name**, **role\_guid**, **user\_name**, and **user\_guid**.

```

select distinct
fff.function_name "function_name",
fsr.role_name "role_name",
fsr.role_guid "role_guid",
fs.user_name "user_name",
fs.user_guid "user_guid"
from
fnd_objects fo,
fnd_form_functions fff,
fnd_grants fg,
fnd_session_roles fsr,
fnd_sessions fs
where

```

```

fo.obj_name = 'XLA_TRANSACTION_ENTITIES' and

fo.object_id = fff.object_id and

fff.object_id = fg.object_id and

fg.role_name = fsr.role_name and

fsr.session_id = fs.session_id and

fs.user_name = &userName and

fg.role_name = &RoleName

when prompted with userName input: sample data is
'AP_SUPV_OPERATIONS'; when prompted with RoleName, sample input is
'AP_ACCOUNTS_PAYABLE_SUPERVISOR_VISION_OPERATIONS_DATA';

```

Sample PL/SQL procedure to initialize the session. Replace **role\_name**, **role\_guid**, **user\_name**, **user\_guid** with results from the previous query.

```

SET SERVEROUTPUT ON SIZE 100000;

DECLARE

l_roleguids FND_TABLE_OF_VARCHAR2_4000 := FND_TABLE_OF_VARCHAR2_4000();
l_rolenames FND_TABLE_OF_VARCHAR2_4000 := FND_TABLE_OF_VARCHAR2_4000();

BEGIN

l_roleguids.extend(1);
l_rolenames.extend(1);
l_roleguids(1) := &role_guid;
l_rolenames(1) := &role_name;

if NVL(fnd_global.session_id, '-1') = '-1' then

fnd_global.initialize_session('X', 'X', l_roleguids, l_rolenames);

end if;

end;

When prompted to enter role_guid: sample input is like
'CB124ED92D967C2233CFD589C0AB110F'
when prompted to enter role_name: sample input is
'AP_ACCOUNTS_PAYABLE_MANAGER_VISION_OPERATIONS_DATA

```

## Updating Event APIs: Examples

The `Update_Event_Status()` API updates the event statuses of more than one event of a transaction. There is a set of overloaded APIs that can be used to update more than one attribute for an event. These APIs update accounting event type, event date, event status, event number, and reference information for an event. All of these API's use the `Update_Event()` API.

---

### Note

Though these update event APIs retain the same name `Update_Event()`, they use the PL/SQL feature of overloading to create unique procedures. In overloading, the input parameter names and types are distinct, resulting in unique procedures. Different columns are updated in the tables, depending on which procedure is called.

---

An event can be updated as long as it is not processed. Once an event is processed, you cannot update the event or the data associated with it.

This topic provides examples of the APIs that update events.

### 1. XLA\_EVENTS\_PUB\_PKG.UPDATE\_EVENT\_STATUS()

In this example, the API updates the event status of one or more events within a transaction matching the specified criteria.

This API checks all source information for valid application, legal entity, event process category, and source IDs. It ensures that the required parameters are not null. The API also validates event status and if passed, event class and event type. The event date is truncated.

```
PROCEDURE XLA_EVENTS_PUB_PKG.update_event_status
(p_event_source_info IN xla_events_pub_pkg.t_event_source_info
,p_event_class_code IN VARCHAR2 DEFAULT NULL
,p_event_type_code IN VARCHAR2 DEFAULT NULL
,p_event_date IN DATE DEFAULT NULL
,p_event_status_code IN VARCHAR2
,p_valuation_method IN VARCHAR2
,p_security_context IN xla_events_pub_pkg.t_security);
```

### 2. XLA\_EVENTS\_PUB\_PKG.UPDATE\_EVENT()

In this example the API updates multiple attributes of a single event. Using this API, the calling program can update event type, event date, and event status. An error code is returned if the update fails.

This API checks all source information for valid application, legal entity, event entity, and source IDs. The API ensures that the event ID is not null, that it belongs to the same transaction as the other transaction information being passed, and that the event has not already been accounted.

The parameters event type, event date, and event status are also validated if passed Not Null.

```
PROCEDURE XLA_EVENTS_PUB_PKG.update_event
(p_event_source_info IN xla_events_pub_pkg.t_event_source_info
,p_event_id IN NUMBER
,p_event_type_code IN VARCHAR2 DEFAULT NULL
,p_event_date IN DATE DEFAULT NULL
,p_event_status_code IN VARCHAR2 DEFAULT NULL
,p_event_number IN NUMBER
,p_reference_info IN
xla_events_pub_pkg.t_event_reference_info
,p_valuation_method IN VARCHAR2
,p_security_context IN xla_events_pub_pkg.t_security);
(p_transaction_date IN DATE DEFAULT NULL)
```

### 3. XLA\_EVENTS\_PUB\_PKG.UPDATE\_EVENT()

In this example the API updates multiple attributes of a single event. Using this API, the calling program can update event type, event date, event status, and event number. An error code is returned if the update fails.

This API checks all source information for valid application, legal entity, event entity, and source IDs. No validations are performed against the event number but if no event number is passed, the routine populates the next highest event number for that transaction. The event date is truncated.

The API ensures that the event ID is not null, that it belongs to the same transaction as the other transaction information being passed, and that the event has not already been accounted. The parameters event type, event date, and event status are also validated if passed Not Null.

```
PROCEDURE XLA_EVENTS_PUB_PKG.UPDATE_EVENT
(p_event_source_info IN xla_events_pub_pkg.t_event_source_info
,p_event_id IN NUMBER
,p_event_type_code IN VARCHAR2 DEFAULT NULL
,p_event_date IN DATE DEFAULT NULL
,p_event_status_code IN VARCHAR2 DEFAULT NULL
,p_event_number IN NUMBER
,p_valuation_method IN VARCHAR2
,p_transaction_date IN DATE DEFAULT NULL
,p_security_context IN xla_events_pub_pkg.t_security);
```

#### 4. XLA\_EVENTS\_PUB\_PKG.UPDATE\_EVENT()

In this example the API updates multiple attributes of a single event. Using this API, the calling program can update event type, event date, event status, and the event's reference information. An error code is returned if the update fails.

This API checks all source information for valid application, legal entity, event entity, and source IDs. No validations are performed on the reference information.

The API ensures that the event ID is not null, that it belongs to the same transaction as the other transaction information being passed, and that the event has not already been accounted. The parameters event type, event date, and event status are also validated if passed Not Null.

```
(p_event_source_info IN xla_events_pub_pkg.t_event_source_info
,p_event_id IN NUMBER
,p_event_type_code IN VARCHAR2 DEFAULT NULL
,p_event_date IN DATE DEFAULT NULL
,p_event_status_code IN VARCHAR2 DEFAULT NULL
,p_reference_info IN xla_events_pub_pkg.t_event_reference_info
,p_valuation_method IN VARCHAR2
,p_transaction_date IN DATE DEFAULT NULL
,p_security_context IN xla_events_pub_pkg.t_security);
```

#### 5. XLA\_EVENTS\_PUB\_PKG.UPDATE\_EVENT()

In this example the API updates multiple attributes of a single event. Using this API, the calling program can update event type, event date, event status, event number, and the event's reference information. An error code is returned if the update fails.

---

#### Note

This API updates both the event's event number and reference information.

---

This API checks all source information for valid application, legal entity, event entity, and source IDs. The API ensures that the event ID is not null, that it belongs to the same transaction as the other transaction information being passed, and that the event has not already been accounted.

The parameters event type, event date, and event status are also validated if passed Not Null. No validations are performed against the event number and reference information, but if no event number is passed, the routine populates the next highest event number for that transaction.

```

PROCEDURE XLA_EVENTS_PUB_PKG.update_event
(p_event_source_info IN xla_events_pub_pkg.t_event_source_info
,p_event_id IN NUMBER
,p_event_type_code IN VARCHAR2 DEFAULT NULL
,p_event_date IN DATE DEFAULT NULL
,p_event_status_code IN VARCHAR2 DEFAULT NULL
,p_valuation_method IN VARCHAR2
,p_transaction_date IN DATE DEFAULT NULL)
,p_security_context IN xla_events_pub_pkg.t_security);

```

## 6. XLA\_EVENTS\_PUB\_PKG.UPDATE\_BULK\_EVENT\_STATUSES

This API updates the event status of multiple events. Before calling this API, users must populate the XLA\_EVENTS\_INT\_GT table with the following:

- application\_id
- entity\_code
- ledger\_id
- event\_id
- event\_status\_code

This API updates the events in the XLA\_EVENTS table to the new status.

This API validates the application ID, event entity, event ID, and event status. The status of both the new and old status cannot be Processed. The new status must be a valid event status.

```

Procedure XLA_EVENTS_PUB_PKG.update_bulk_event_statuses
(p_application_id IN INTEGER);

```

## Deleting Event APIs: Examples

This topic provides examples of the APIs that delete events.

### 1. XLA\_EVENTS\_PUB\_PKG.DELETE\_EVENT()

This API deletes an unaccounted event based on its event identifier. The API returns an error code if the delete fails.

The API checks all source information for valid application, legal entity, event process category, and source IDs. The API also ensures that the mandatory parameters are not null, that the event ID belongs to the same transaction as the other transaction information being passed, and that the event has not been accounted.

```

PROCEDURE XLA_EVENTS_PUB_PKG.delete_event
(p_event_source_info IN xla_events_pub_pkg.t_event_source_info
,p_event_id IN NUMBER
,p_valuation_method IN VARCHAR2
,p_security_context IN xla_events_pub_pkg.t_security);

```

### 2. XLA\_EVENTS\_PUB\_PKG.DELETE\_EVENTS()

This API deletes all events for a transaction that meet the specified criteria. When called, events that belong to the given accounting event class, event type, and event date are deleted.

The API checks all source information for valid application, legal entity, event process category, and source IDs. It ensures that the required parameters are not null and if passed, validates the event type and event status.

The function returns the number of events deleted.

```
FUNCTION XLA_EVENTS_PUB_PKG.delete_events
(p_event_source_info IN xla_events_pub_pkg.t_event_source_info
,p_event_class_code IN VARCHAR2 DEFAULT NULL
,p_event_type_code IN VARCHAR2 DEFAULT NULL
,p_event_date IN DATE DEFAULT NULL
,p_valuation_method IN VARCHAR2
,p_security_context IN xla_events_pub_pkg.t_security)
RETURN INTEGER;
```

### 3. XLA\_EVENTS\_PUB\_PKG.DELETE\_ENTITY()

This API deletes a row from the XLA\_TRANSACTION\_ENTITIES table. The routine checks if there are still events associated with the transaction. If yes, the routine does nothing and returns 1; otherwise, it deletes the transaction in the XLA\_TRANSACTION\_ENTITIES table and returns 0.

There are no validations for this API.

If a transaction is deleted, users must call the DELETE\_ENTITY API to delete the row in the XLA\_TRANSACTION\_ENTITIES table.

```
Function XLA_EVENTS_PUB_PKG.delete_entity
(p_source_info IN xla_events_pub_pkg.t_event_source_info
,p_valuation_method IN VARCHAR2
,p_security_context IN xla_events_pub_pkg.t_security)
RETURN INTEGER;
```

### 4. XLA\_EVENTS\_PUB\_PKG.DELETE\_BULK\_EVENTS()

This API deletes multiple events. Before calling this API, users must populate the XLA\_EVENTS\_INT\_GT table with the following:

- application\_id
- entity\_code
- ledger\_id
- event\_id
- event\_status\_code

The API deletes events from the XLA\_EVENTS table.

This API validates the application ID, event process category, and event ID. The status of the event to be deleted cannot be processed.

```
Procedure XLA_EVENTS_PUB_PKG.delete_bulk_events
(p_application_id IN INTEGER);
```

## Common Parameters: Explained

This topic provides details on the parameters common to many event APIs.

This topic includes information on the following types of parameters:

- Transaction identifiers
- Contextual information
- Transaction security identifiers
- Transaction reference information
- Event information

### Transaction Identifiers

The table below describes transaction identifier attributes.

Parameter Name	Type	Description
p_transaction_number	Varchar2(240)	Transaction number of the event-based transaction. This is the user transaction identifier and serves as a reference for the transaction.
p_event_source_info	xla_events_pub_pkg.t_event_source_info	System transaction identifiers

### Contextual Information

The table below describes contextual information details.

Parameter Name	Type	Description
p_source_application_id	Integer	Internal identifier of the application that generates the document or transaction. This may be different from the application that generates and/or owns the accounting for the corresponding event.  If no value is provided, the default is p_application_id.  Used for Oracle subledgers. This parameter is unlikely to be used for non-Oracle systems.
p_application_id	Number	Application internal identifier.
p_legal_entity_id	Number	Legal entity internal identifier.
p_ledger_id	Number	Ledger internal identifier.
p_valuation_method	Varchar2(30)	Valuation method used for securing a transaction. Some applications secure their transactions by valuation method.  Used for Oracle subledgers. This parameter is unlikely to be used for non-Oracle systems.

## Transaction Security Identifiers

The table below describes transaction security identifier attributes.

Parameter Name	Type	Description
p_security_context	xla_events_pub_pkg.t_Security	Security context information for the transaction that has created the events.

## Transaction Reference Information

The table below describes transaction reference parameter attributes.

Parameter Name	Type	Description
p_array_reference_info	xla_events_pub_pkg.t_array_event_reference_info	Array of optional reference information for multiple events. These are stored with the events.
p_reference_info	xla_events_pub_pkg.t_event_reference_info	Optional reference information for a particular event.

## Event Information

The table below describes event information parameter attributes.

Parameter Name	Type	Description
p_entity_type_code	Varchar2(30)	Entity type internal code.
p_event_class_code	Varchar2(30)	Event class internal code.
p_event_type_code	Varchar2(30)	Event type internal code
p_event_id	Number	Event internal identifier.
p_event_date	Date	Event accounting date.
p_event_status_code	Varchar2(1)	External status code for an event.
p_event_number	Number	Event sequence number within a transaction. Events are ordered by this sequence number for accounting.
p_array_entity_source_info	xla_events_pub_pkg.t_array_entity_source_info	Array of transaction source ID information as stamped on the entity.
p_array_event_type_code	xla_events_pub_pkg.t_array_event_type_code	Array of internal codes for the event type as defined by applications.
p_array_event_date	xla_events_pub_pkg.t_array_event_date	Array of accounting dates for events.
p_array_event_status_code	xla_events_pub_pkg.t_array_event_status_code	Array of external status codes for events.
p_array_event_number	xla_events_pub_pkg.t_array_event_number	Array of event sequence numbers within a transaction. Events are ordered by these sequence numbers for accounting.

p_array_entity_event_info	xla_events_pub_pkg.t_array_entity_event_info	Array of combined entity and event attributes.
---------------------------	--	--

## PL/SQL Data Types: Explained

The following are the predefined PL/SQL data structures available in the XLA\_EVENTS\_PUB\_PKG package.

### PL/SQL Record Structures

#### Transaction Source Information

```
TYPE t_event_source_info IS RECORD
(source_application_id NUMBER DEFAULT NULL
, application_id NUMBER
, legal_entity_id NUMBER
, ledger_id NUMBER
, entity_type_code VARCHAR2(30)
, transaction_number VARCHAR2(240)
, source_id_int_1 NUMBER
, source_id_int_2 NUMBER
, source_id_int_3 NUMBER
, source_id_int_4 NUMBER
, source_id_char_1 VARCHAR2(30)
, source_id_char_2 VARCHAR2(30)
, source_id_char_3 VARCHAR2(30)
, source_id_char_4 VARCHAR2(30));
```

The table below provides descriptions on select attributes listed above.

Attribute Table #1

Attribute	Description
source_application_id	Internal identifier of the application that generates the document or transaction. This may be different from the application that generates and or owns the accounting for the corresponding event. Source applications do not need to be registered as subledger applications.  If no value is provided, the default is application_id.  Used for Oracle subledgers. This parameter is unlikely to be used for non-Oracle systems.
application_id	Application transaction owner identifier.
legal_entity_id	Transaction legal entity identifier.
ledger_id	Transaction ledger identifier.
entity_type_code	Entity code as defined by applications during setup.
transaction_number	Transaction number of the transaction that has created the events. The transaction number serves as a reference for the transaction.
source_id_xxx_n	Generic columns that store the identifier for the transaction in the transaction table.

### Event Reference Information

```

TYPE t_event_reference_info IS RECORD
(reference_num_1 NUMBER
,reference_num_2 NUMBER
,reference_num_3 NUMBER
,reference_num_4 NUMBER
,reference_char_1 VARCHAR2(240)
,reference_char_2 VARCHAR2(240)
,reference_char_3 VARCHAR2(240)
,reference_char_4 VARCHAR2(240)
,reference_date_1 DATE
,reference_date_2 DATE
,reference_date_3 DATE
,reference_date_4 DATE);

```

---

### Note

See Attribute Table #2 for descriptions on select attributes.

---

### Event Information

```

TYPE t_event_info IS RECORD
(event_id NUMBER
,event_number NUMBER
,event_type_code VARCHAR2(30)
,event_date DATE
,event_status_code VARCHAR2(1)
,process_status_code VARCHAR2(1)
,reference_num_1 NUMBER
,reference_num_2 NUMBER
,reference_num_3 NUMBER
,reference_num_4 NUMBER
,reference_char_1 VARCHAR2(240)
,reference_char_2 VARCHAR2(240)
,reference_char_3 VARCHAR2(240)
,reference_char_4 VARCHAR2(240)
,reference_date_1 DATE
,reference_date_2 DATE
,reference_date_3 DATE
,reference_date_4 DATE);

```

---

### Note

See Attribute Table #2 for descriptions on select attributes.

---

The table below provides descriptions on select attributes listed above.

Attribute Table #2

Attribute	Description
event_type_code	Code for the event type of the event, as defined during setup.
event_date	Event accounting date.
event_id	Event internal identifier.
event_number	Event sequence number for the event within a transaction. Events are processed in the order of their event number.
event_status_code	Status code for the event. This is the event's external status and is used by implementers.

transaction_number	Transaction number of the transaction that has created the events. The transaction number serves as a reference for the transaction.
source_id_xxx_n	Generic columns that store the identifier for the transaction in the transaction table.
reference_xxx_n	Generic columns that store reference information for the event.
valuation_method	Valuation method code used as a security context for applications that support the valuation method.  Used for Oracle subledgers. This parameter is unlikely to be used for non-Oracle systems.
security_id_xxx_n	Security contexts.

### Security Context Information

Security context information is optional and restricts a user's access to several features in the Oracle Fusion Accounting Hub. In conjunction with the Fusion Security model, the security context information will determine:

- Which events the Create Accounting process will process.
- Which events and entries will be visible to the user on the screens and reports.

If used, the securing organization's IDs should be populated in the attributes. As an example, a securing organization can be a business unit or a ledger.

Use the following record structure to pass security context information through event APIs. This structure is defined in XLA\_EVENTS\_PUB\_PKG package.

```
TYPE t_security IS RECORD
(security_id_int_1 NUMBER
,security_id_int_2 NUMBER
,security_id_int_3 NUMBER
,security_id_char_1 VARCHAR2(30)
,security_id_char_2 VARCHAR2(30)
,security_id_char_3 VARCHAR2(30));
```

The table below provides descriptions for the attributes listed above.

Attribute Table #3

Attribute	Description
security_id_int_n	Security context information in INTEGER type.
security_id_char_n	Security context information in VARCHAR type.

### PL/SQL Table Structure

Array of Information based on above Structures

```
TYPE t_array_event_reference_info IS TABLE OF t_event_reference_info
TYPE t_array_event_info IS TABLE OF t_event_info
TYPE t_array_event_source_info IS TABLE OF t_event_source_info
```

Other Array Structures

```

TYPE t_array_event_type IS TABLE OF VARCHAR2(30)

TYPE t_array_event_date IS TABLE OF DATE
TYPE t_array_event_status_code IS TABLE OF VARCHAR2(1)
TYPE t_array_entity_id IS TABLE OF NUMBER
TYPE t_array_event_id IS TABLE OF NUMBER
TYPE t_array_event_number IS TABLE OF NUMBER

```

## Constants: Explained

The Oracle Fusion Accounting Hub uses predefined accounting event status Constants. These constants are defined in the XLA\_EVENTS\_PUB\_PKG package.

### Event Statuses and Constants

The table below lists event statuses and the corresponding constants that must be used when employing event APIs.

Event Status	Constant
Incomplete	XLA_EVENTS_PUB_PKG.C_EVENT_INCOMPLETE
Unprocessed	XLA_EVENTS_PUB_PKG.C_EVENT_UNPROCESSED
No Action	XLA_EVENTS_PUB_PKG.C_EVENT_NOACTION
Processed	XLA_EVENTS_PUB_PKG.C_EVENT_PROCESSED

## Create Accounting Integration

### Create Accounting Integration: Overview

The Oracle Fusion Accounting Hub enables customers to extend and customize the integration with the Create Accounting process by invoking API calls. The Accounting Hub code structure enables the integration of customized business logic at various stages during the accounting process.

Specifically, you will need to customize the xla\_acct\_hooks\_pkg if you want to add custom logic.

Implementers of the Accounting Hub at the customer site can add logic to these APIs that they want the Create Accounting process to perform at the indicated step.

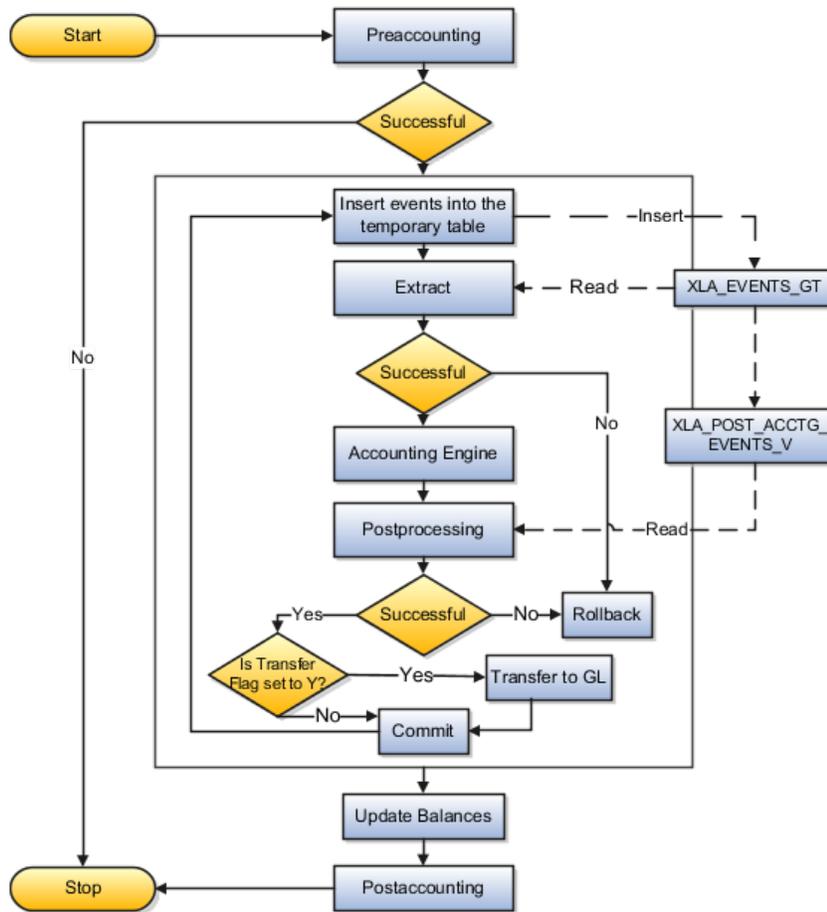
- Preaccounting
- Create Accounting extract
- Postprocessing
- Postaccounting

### Timing and Positioning of APIs: Explained

This section describes the life cycle of the Create Accounting process.

## Create Accounting Process Life Cycle

The figure below shows the life cycle of the Create Accounting process.



### Preaccounting

The preaccounting logic is called at the beginning of the Create Accounting process before selecting accounting events for processing.

Example: This can be used to evaluate a condition to identify transactions for processing.

### Extract

The extract logic is called for all modes of accounting, final and draft, in batch as well as document mode. It is called after selecting the next set of events to be processed, but before any processing is done. Events selected for processing are made available to custom logic through the global temporary table XLA\_EVENTS\_GT. Implementers adding code to this API can use it to populate transaction objects based on information in the XLA\_EVENTS\_GT table.

Example: Use this to populate transaction objects.

### Postprocessing

The postprocessing logic is called after creating subledger journal entries for each processing unit (commit unit). Custom logic within this API can use the view

XLA\_POST\_ACCTG\_EVENTS\_V to determine which accounting events were successfully accounted by the Create Accounting process.

Example: Use this to update posted flags on the transaction distributions to indicate whether the transaction was accounted successfully.

### Postaccounting

The postaccounting logic is called after subledger journal entries are successfully committed in the database.

Examples: Use this to send a notification to a system administrator that the accounting process has completed.

Or, use this to export the supporting reference balances to a third party reconciliation tool for comparison to source system totals to ensure accounting is complete.

### XLA\_POST\_ACCTG\_EVENTS\_V

The table below describes the view structure.

Column Name	Null?	Data Type	Description
APPLICATION_ID	Not Null	NUMBER(18)	Internal identifier for the application to which the event belongs
ENTITY_CODE	Not Null	VARCHAR2(30)	Event entity type code
ENTITY_ID	Not Null	NUMBER(18)	Internal identifier for the entity representing the actual document
EVENT_CLASS_CODE	Not Null	VARCHAR2(30)	Event class code
EVENT_DATE	Not Null	DATE	Event or accounting date for the event
EVENT_ID	Not Null	NUMBER(18)	Event internal identifier
EVENT_NUMBER	Not Null	NUMBER(18)	Event number assigned to the event within the document
EVENT_STATUS_CODE	Not Null	VARCHAR2(1)	Event status code
EVENT_TYPE_CODE	Null	VARCHAR2(30)	Code for the event type that classifies the event being created
LEDGER_ID	Not Null	NUMBER(18)	Ledger internal identifier to which event the belongs
LEGAL_ENTITY_ID	Null	NUMBER(18)	Internal identifier for the legal entity
PROCESS_STATUS_CODE	Not Null	VARCHAR2(1)	Event process code
REFERENCE_NUM_1	Null	NUMBER	Reference information
REFERENCE_NUM_2	Null	NUMBER	Reference information
REFERENCE_NUM_3	Null	NUMBER	Reference information
REFERENCE_NUM_4	Null	NUMBER	Reference information

REFERENCE_CHAR_1	Null	VARCHAR2(240)	Reference information
REFERENCE_CHAR_2	Null	VARCHAR2(240)	Reference information
REFERENCE_CHAR_3	Null	VARCHAR2(240)	Reference information
REFERENCE_CHAR_4	Null	VARCHAR2(240)	Reference information
REFERENCE_DATE_1	Null	DATE	Reference information
REFERENCE_DATE_2	Null	DATE	Reference information
REFERENCE_DATE_3	Null	DATE	Reference information
REFERENCE_DATE_4	Null	DATE	Reference information
SOURCE_ID_INT_1	Null	NUMBER(18)	Placeholder column that stores internal identifier of the document being represented by the entity
SOURCE_ID_INT_2	Null	NUMBER(18)	Placeholder column that stores internal identifier of the document being represented by the entity
SOURCE_ID_INT_3	Null	NUMBER(18)	Placeholder column that stores internal identifier of the document being represented by the entity
SOURCE_ID_INT_4	Null	NUMBER(18)	Placeholder column that stores internal identifier of the document being represented by the entity
SOURCE_ID_CHAR_1	Null	VARCHAR2(30)	Placeholder column that stores internal identifier of the document being represented by the entity
SOURCE_ID_CHAR_2	Null	VARCHAR2(30)	Placeholder column that stores internal identifier of the document being represented by the entity
SOURCE_ID_CHAR_3	Null	VARCHAR2(30)	Placeholder column that stores internal identifier of the document being represented by the entity
SOURCE_ID_CHAR_4	Null	VARCHAR2(30)	Placeholder column that stores internal identifier of the document being represented by the entity
TRANSACTION_NUMBER	Null	VARCHAR2(240)	Transaction number given to the document by the products owning the document

## Parameter Specifications: Explained

This section describes parameters for the Create Accounting process integration points.

## Preaccounting Parameters

The table below describes the parameters for xla\_acct\_hooks\_pkg.preaccounting().

Parameter Name	Description
P_APPLICATION_ID	Application identifier for which the Create Accounting process is submitted. The custom logic checks this parameter first to see if it is the desired application.
P_LEDGER_ID	Ledger identifier for which the Create Accounting process is submitted.
P_PROCESS_CATEGORY	Process category specified by you when launching the Create Accounting process.
P_END_DATE	End date specified by you.
P_ACCOUNTING_MODE	Indicates the mode in which the Create Accounting process is submitted.
P_REPORT_REQUEST_ID	Request ID of the Create Accounting process submitted.
P_VALUATION_METHOD	Valuation method specified by you. Relevant for internal Oracle applications.
P_SECURITY_ID_INT_1 P_SECURITY_ID_INT_2 P_SECURITY_ID_INT_3 P_SECURITY_ID_CHAR_1 P_SECURITY_ID_CHAR_2 P_SECURITY_ID_CHAR_3	Security context values as passed in as a parameter for the Create Accounting process.

## Extract Parameters

The table below describes the parameters for xla\_acct\_hooks\_pkg.extract().

Parameter Name	Description
P_APPLICATION_ID	Application identifier for which the Create Accounting process is submitted. The custom logic checks this parameter first to see if it is the desired application.
P_ACCOUNTING_MODE	Indicates the mode in which the Create Accounting process is submitted

## Postprocessing Parameters

The table that describes the parameters for xla\_acct\_hooks\_pkg.postprocessing() is the same as the table that defines the parameters for extract.

## Postaccounting Parameters

The table that describes the parameters for xla\_acct\_hooks\_pkg.postaccounting() is the same as the table that defines the parameters for preaccounting.

## FAQs for Create Account Integration

### How can I integrate non-Oracle applications or customizations with the Oracle Fusion Accounting Hub?

To integrate non-Oracle applications or customizations with the Oracle Fusion Accounting Hub, you can add logic to the preaccounting, extract, postprocessing, or postaccounting APIs in the `xla_acct_hook_pkg`.

In order to improve the performance, it is recommended that you check the Application ID as the first step in your logic and exit if it does not match.

### What happens if there is an exception to the API logic?

If the API logic raises an exception, then that exception will abort the Create Accounting process.

If the logic that raises the exception is in preaccounting, then the Create Accounting process will immediately rollback and exit without doing any accounting.

If the logic that raises the exception is in extract or postprocessing, then any accounting done for the current set of transactions will be aborted and rolled back, and the worker will exit. However, other workers may continue processing.

If the logic that raises the exception is in postaccounting, then the Create Accounting process will rollback and exit. However, this will only rollback anything done by the postaccounting logic. It will not rollback any accounting done by the Create Accounting process.

### Can I receive a notification of the status of a Create Accounting process?

In Oracle Fusion Accounting Hub you can add logic to the postaccounting API to notify you of the status of the Create Accounting process.

## Manage Accounting Rules

### Creating Accounting Method: Explained

Accounting methods group subledger journal entry rule sets together to define a consistent accounting treatment for each of the accounting event classes and accounting event types for all subledger applications. The grouping allows a set of subledger journal entry rule sets to be assigned collectively to a ledger.

For example, a subledger accounting method entitled US GAAP can be defined to group subledger journal entry rule sets that adhere to and comply with US Generally Accepted Accounting Principles (GAAP) criteria.

By assigning a different subledger accounting method to each related ledger, you can create multiple accounting representations of transactions.

Accounting rules can be defined with either a top down, or a bottom up approach. When defining subledger accounting rules from the top down, you will initially define the accounting method followed by components of each rule, which will need to be assigned to it. When defining subledger accounting rules from the bottom up, you will initially define components for each rule and then assign them as required.

The Create Accounting process uses the accounting method definition with active journal entry rule set assignments to create subledger journal entries.

When an accounting method is initially defined, or after modifying a component of any accounting rule associated to the assigned journal entry rule set, its status changes to Incomplete.

The accounting method must be completed, by activating its journal entry rule set assignments, so that it can be used to create accounting.

The following definitions are utilized to define the journal entries, and are applied as updates to the accounting method:

- Updates to the predefined accounting method
- Assignment of journal entry rule sets for an accounting event class and/or accounting event type from the accounting methods page
- Assignment of accounting methods to ledgers
- Activation of subledger journal entry rule set assignments

### **Updates on Predefined Accounting Method**

You may update a predefined accounting method by end dating the existing assignment and creating a new assignment with an effective start date.

### **Assignment of Journal Entry Rule Set for Accounting Event Class and Accounting Event Type**

You create the assignment of a journal entry rule set for an accounting event class and accounting event type using the accounting method page.

The following should be considered for assigning rule sets:

- If the accounting method has an assigned chart of accounts, you can select journal entry rule sets that use that same chart of accounts, or that are not associated with any chart of accounts.
- Select an option to assign existing journal entry rule sets or define a new one.

### **Assignment of Accounting Methods to Ledgers**

If the accounting method has an assigned chart of accounts, it may only be used by ledgers that use the same chart of accounts.

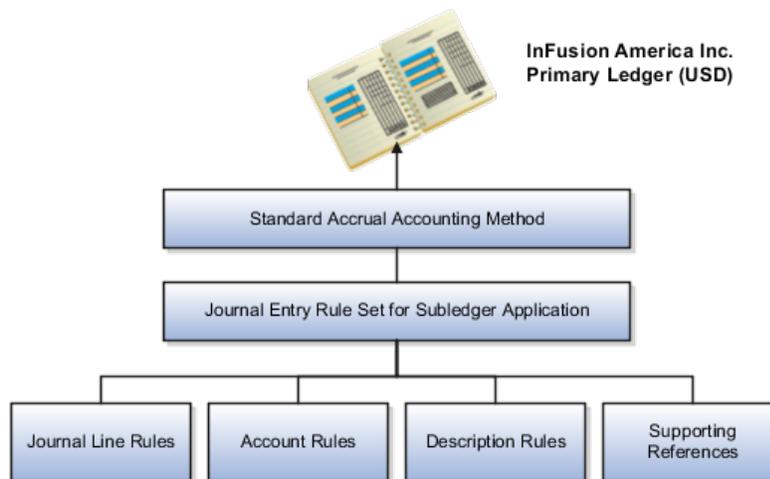
If the accounting method does not have an assigned chart of accounts, the accounting method can be assigned to any ledger.

## Activation of Subledger Journal Entry Rule Set Assignments

You can activate the subledger journal entry rule set assignments from the Accounting Method page. You can also submit the Activate Subledger Journal Entry Rule Set Assignments process to validate and activate your accounting setups.

## Fusion Setup Flow

The figure below shows the relationship of components making up an accounting method as described in the above text.



## Creating Subledger Journal Entry Rule Sets: Explained

Subledger journal entry rule sets provide the definition for generating a complete journal entry for an accounting event.

Select the option to define the subledger journal entry rule set for a particular accounting event class or accounting event type.

If you are using multiple ledgers to meet divergent and mutually exclusive accounting requirements, you can vary journal entry rule sets by ledger. Each of the subledger journal entry rule sets can meet a specific type of accounting requirements.

For example, use US Generally Accepted Accounting Principles (GAAP) oriented subledger journal entry rule sets for a ledger dedicated to US GAAP reporting, and French statutory accounting conventions for a ledger dedicated to French statutory reporting. These two sets of definitions have differences based on the setup of the various components that make up their subledger journal entry rule sets.

Seeded subledger journal entry rule sets are provided for all Oracle subledgers. If specific requirements are not met by seeded subledger journal entry rule sets, users can create new ones of copy the seeded definitions and then rename and modify the new copied definitions and their assignments.

Subledger journal entry rule set assignments can be made at two levels, header and line. The following are the subcomponents of a subledger journal entry rule set:

- Description rules
- Journal line rules
- Account rules
- Supporting references

### **Assignment at Header Level**

Header assignments define subledger journal header information and line assignments define journal line accounting treatment.

A header assignment includes the following:

- Accounting date (required)
- Accrual reversal accounting date (optional)
- Description rule (optional)
- Supporting references (optional)

### **Assignment at Line Level**

You can define multiple subledger journal entry rule sets for an accounting event class or accounting event type. A single journal entry is generated per accounting event per ledger using the line assignments from the journal entry rule set assigned to the accounting event class or accounting event type.

The following can be assigned to a journal entry line:

- Journal line description rule
- Journal line rule
- Account rule
- Supporting references

### **Assignment of Description Rules**

If a description rule is defined with sources, the sources must also be assigned to the accounting event class that is assigned to the journal entry rule set. The description rule may be assigned at either the header or line level of the journal entry or to both levels.

### **Assignment of Journal Line Rules**

When assigning the journal line rule, you must identify the line type: Gain, Loss, Gain or Loss, Credit, or Debit. The journal line rule must be assigned to the same accounting event class as the one assigned to the subledger journal entry rule set.

When assigning a journal line rule that is enabled for accounting for a business flow, the account combination and certain accounting attribute values are copied from its related journal line having the same business flow class as the current line. Optionally, copy the description rule into the current line instead of assigning a separate description rule.

When assigning a journal line rule that is enabled to copy from the corresponding line within the same journal entry, you have the option to copy the account combination, the segment value, or the line description from the corresponding line into the current line.

### **Assignment of Account Rules**

The account rule assignment will define which accounts will be used for the subledger journal line. If the account rule is setup with a chart of accounts, it must have the same chart of accounts as the one assigned to the journal entry rule set. When account rules are defined with sources, the sources must also be assigned to the accounting event class that is assigned the journal entry rule set.

There are two types of account rules:

- Account Combination Rule: Assign an account combination rule to derive the account combination.
- Segment Rule: Assign a segment rule to derive a specific segment of an account. For example, a cost center or a natural account segment.

### **Assignment of Supporting References**

Supporting references may be assigned at the header or line level of the journal entry to capture transaction values on the journal entry header or lines. If the supporting reference segments are assigned multiple sources, at least one source must also be assigned to the accounting event class that is assigned the journal entry rule set.

## **Journal Line Rules: Explained**

Journal line rules are defined within the context of accounting event classes. A journal line rule can be used in a subledger journal entry rule set that has the same event class. You may also assign conditions to the journal line rule.

### **Journal Line Rules**

Journal line rules are assigned to journal entry rule sets.

To create a journal line rule, select values for options such as:

- Side (Debit, Credit, Gain or Loss)

For example, when an Oracle Fusion Payables invoice is generated, the liability account should normally be credited. The journal line rule must therefore specify the Side option as Credit. On the other hand, the payment of the Payables invoice must be accounted with a debit to the

liability account. A separate journal line rule must be defined to create this debit line.

- **Merge Matching Lines:** To summarize subledger journal entry lines within each subledger entry. Journal entry lines with matching criteria are merged.
- **Accounting Class**
  - Select an accounting class to classify journal entry lines.
  - For example, when a validated Payables invoice is accounted, the Item Expense and Liability journal lines are created. In this case, the journal line rules used in the accounting rules are assigned Item Expense and Liability accounting classes respectively.
- **Conditions:** To restrict the use of a journal line rule by controlling when a particular journal line rule is used by the Create Accounting process.
- **Accounting Attributes:** When creating a journal line rule, accounting attribute assignments are automatically established based on the default accounting attribute assignments for that journal line rule's accounting event class. You can override this default mapping of standard sources to accounting attributes. The list of values for the source override includes all sources assigned to the accounting attribute for the event class associated with the journal line rule.
- **Advanced Options**
  - **The Subledger Gain or Less Option:** Applies only to amount calculations for the primary ledger. Gain or loss amounts are not converted to reporting currency or non-valuation method secondary ledgers. If the option is selected, the journal line holds the gain or loss amounts calculated by the subledger.

The gain or loss amount is calculated as the difference in applied amounts due to fluctuations in exchange rates based upon conversion to the ledger currency. Foreign exchange gain or loss amounts occur when two related transactions, such as an invoice and its payment, are entered in a currency other than the ledger currency, and the conversion rate fluctuates between the times that the two are accounted.

- **The Rounding Class Option:** Along with the transaction rounding reference group journal lines together and calculates transaction rounding. Subledger transaction rounding differences can occur when a transaction has multiple related applied-to transactions, such as when a Receivables invoice has multiple associated receipts.
- **The Link Journal Lines Option:** Determines whether the journal line rule is set up to establish a link between the accounting of transactions that are related both within the same application, and across applications. The alternatives are described in this table:

Link Journal Lines Option	Description
None	No link is established.

Copy from corresponding line	Build account for a journal line using segments from the offsetting entry of the current journal line. For example, when the business process requires that a cost center incurring an expense must also bear the invoice liability and cash outlay.
Business flow	Link logically related business transactions. For example, when recording the closing of a loan, you can link to the account that was used to book the loan origination. Journal line rules that are linked must also be assigned the same business flow class.

## Defining Conditions for Journal Line Rules

You may set conditions to specify whether the journal line rule will be used to create a subledger journal entry line. If the conditions are true, the line rule is used to create a subledger journal entry line. Use sources to create these conditions.

For example, you can set up a condition that will create a journal line to record tax, only if there is tax for an invoice. The line type and account class mentioned here are examples of sources.

- The condition for a Payables invoice tax journal line rule could be:
  - Where Line Type = Tax
  - When this condition is true, there is tax for a payables invoice line. A journal entry line is created to record the accounting impact of the tax.
- Similarly, the condition for a Oracle Fusion Receivables invoice tax journal line rule could be:
  - Where Account Class = Tax
  - In this case, if there is an account class of Tax, the journal line is used to record the accounting impact of the tax.

Another example is a condition that creates a journal line for freight when there are freight charges on an invoice.

Journal line rule conditions determine whether a journal line rule and its associated account rules and description rules, are used to create the subledger journal entry line.

---

### Note

Constant values that are used in any Conditions region must not contain the following characters:

- "
- ,
- &
- |
- (

- )
- '

For example, in the condition "Project Type" = ABC (123), the constant value following the equal sign, ABC (123), contains restricted characters ( ) that enclose 123 and is invalid.

---

## Account Rules: Explained

Account rules are used to determine the accounts for subledger journal entry lines. In addition, you can specify the conditions under which these rules apply. Using these capabilities, you can develop complex rules for defining accounts under different circumstances to meet your specific requirements. You can define account rules for an account, segment, or value set.

### Account Rules by Account

Define account rules by account to determine the entire account combination. For example, an account rule defined by account can be used to determine the complete supplier liability account in Oracle Fusion Payables.

### Account Rules by Segment

Define segment rules to derive a specific segment of the general ledger account. For example, a particular segment like the company segment can be determined from the distribution account. Another segment can be determined with the use of a constant value. Creating the account one segment at a time offers greater flexibility, but also requires more setup.

Use both segment based and account based rules to derive a single account. Segment specific rules are used, where they are defined, and take the remaining values from an account based rule. For example, you can select an account rule which is for all segments and also separately select a rule which is for one particular segment. Segment specific rules take precedence over the all segments account based rule.

Combine account rules with segment rules. In this case, the segment value is derived from the segment rule to override the corresponding segment of the account. However, if the segment rule has conditions associated with the priorities and none of the conditions are met, no override occurs and therefore, the segment value is derived from the account rule.

---

### Note

If the returned account is end dated with a date that is the same or before the subledger journal entry accounting date and an alternate account is defined in Oracle Fusion General Ledger, an alternate account is used. The original account is stored on the journal line for audit purposes.

If the alternate account is invalid, and the **Post Invalid Accounts to Suspense Account** option is selected in the Create Accounting process, then a suspense account is used. An error message is displayed if a valid suspense account is not available.

---

## Account Rules by Value Sets

In the absence of a chart of accounts, you may define account rules based upon value sets. This enables you to share the same rule between more than one chart of accounts if the segments in these charts of accounts share the same value set.

## Sharing Account Rules across Applications

You may share account rules across applications in the following ways.

- Assign an account rule from the same or a different application to a journal line rule in the subledger journal entry rule set. For example, to derive an expense account for journal line rule Expense, assign the Projects Cost Account rule owned by Oracle Fusion Projects to the Payables journal line rule Expense.
- Create an account rule based on an account rule from another application and assign it to a journal line rule. For example, you may create a new account rule Invoice Expense Account referencing Project Cost Account assigned in the Priorities region. You may attach the Invoice Expense Account rule to the journal line rule Expense in the journal entry rule set.

---

### Note

To share an account rule across applications, all sources used by the account rule must be available for the event class.

If the sources are available, an account rule is assigned to a journal line rule in the journal entry rule set, and verification occurs to confirm that all sources used by the account rule are available for the journal line rule accounting event class. Journal line rules are only available if the sources are shared; such as reference objects.

---

## Account Rules and Mapping Sets

Mapping sets can be used to associate a specific output value for an account or segment. You can use mapping sets in account rules to build the account.

## Account Rules Conditions

In the account rules you may specify conditions for each rule detail line. Priorities determine the order in which account rule conditions are examined. When the condition is met, the rule associated with that priority is used. Depending on which of the defined conditions is met, a different account rule detail is employed to create the account.

The Create Accounting process evaluates conditions based on the priority of the rule detail. When the condition is met, the rule detail is applied.

## Creating Account Rules: Points to Consider

You can define an account rule using the following rule types:

- Account combination
- Segment
- Value set

## Account Combination Rules

Set up account combination rules based upon the following value types:

1. Source Value Type: Derive the account combination by specifying a source. Sources that have been set up as accounts can be assigned to an account combination rule. Oracle Fusion Subledger Accounting then obtains the code combination identifier from the source.
2. Constant Value Type: Establish the account as a constant value.

For example, the constant could be a completed account combination from the chart of accounts specified. An example is the account combination, 01.000.2210.0000.000. This is the simplest way to derive an account.

3. Mapping Set Value Type: Derive the account combination by referencing a mapping set. Set up a mapping set to determine the complete account combination from the chart of accounts specified.
4. Account Rule Value Type: Derive the account by referencing another account rule.

The chart of accounts does not need to be specified when defining this type of rule. If the account rule has a chart of accounts assigned, then all the related account rules must use the same or no chart of accounts.

---

### Note

A chart of accounts must be specified for rules using constants.

---

## Segment Rules

Set up segment rules as follows:

- When a chart of accounts is specified, create a rule to derive the value for a specific segment from the chart of accounts.
- If the chart of accounts is not specified, create a rule to derive the value for an account segment with a specific qualifier.

Set up segment rules using the same methods discussed in the preceding Account Combination Rules section. By specifying different value types, users can select the way in which the segment value is derived.

---

### Note

A chart of accounts must be specified for rules using constants.

---

## Value Set Rules

Value set based rules can be created when a chart of accounts is not specified. This enables you to share the same rule between more than one chart of accounts if the segments in these charts of accounts share the same value set.

Set up value set based rules using the same methods discussed in the preceding Account Combination Rules section.

## Mapping Sets: Explained

Use mapping sets to quickly define a specific output value with an account or a segment. Based on the input value from subledger transactions or reference information, a specific value can be assigned to a segment or values can be assigned to all segments of the account. Mapping sets provide an efficient way to define the output values and are easier than using the account rule conditions.

To define a mapping set, pairs of values are specified. For each input value, specify a corresponding account combination or segment output value. One or more related pairs of these input values with the segment or account output values form a mapping set. Use value sets or lookup types for validating the input values of the mapping set.

For example, it is possible to create a mapping set based on two input values, Yes and No. Apply these input values to determine the balancing segment value of an account: 01 if the input value is Yes and 02 if the input value is No. Use this mapping set in one of the rules that builds the segment values of an account. The rule compares the value of a source to determine if it is Yes or No and then enters the segment value accordingly.

Another example is where a business has three major regions: East, South, and West. The business has a Region Code segment in their chart of accounts. Region names can be input values in the applications, such as transaction type names and service codes. These input values can be included with other information about the transaction and become part of the source information. Users can create a mapping set that maps region names to the corresponding region code as described in the table below.

Input Value	Segment Value
East	01
South	02
West	03

## Defining Mapping Sets: Examples

Define a mapping set when you have a matrix of input values that produces distinct output values. For each input value, specify a corresponding account combination or segment output value. One or more related pairs of these input

values with the segment or account combination output values form a mapping set.

A mapping set definition includes the selection of input sources, output type, and mappings. The mappings section displays how input values are mapped to output values.

To define mapping sets:

- Specify the output type: The output type for a mapping set can be an account combination, segment, or value set. Use a value set, if the value set is used by more than one chart of accounts, and the definition of the mapping set can be reused across multiple charts of accounts. Then expected input or output combinations are constant across the charts of accounts. Based on the selection, the mapping set provides the value for an account, segment, or value set.
- Define the input source: Specify the input source for mapping. The input source is predefined for seeded mapping sets
- Define the chart of accounts and value sets:

Mapping sets are used with account rules:

- If the output type is account combination or segment, identify the chart of accounts assigned to the mapping set.
- If the output type is a value set, identify the value set assigned to the mapping set.
- If defining a mapping set for more than one chart of accounts or value sets, the same mapping set can be assigned to more than one account rule. This increases share-ability of the mapping set.
- Specify the output value for the mapping: For a given input value, enter the corresponding output value. The account rule uses this value to populate either the account or the segment.
  - If the output type is a value set, the output value is an individual value from the value set entered.
  - If the output type is segment, the output value is an individual segment value.
  - If the output type is account combination, the output value is an entire account.
- A mapping set with no associated chart of accounts:
  - Can be assigned to an account rule, if the account rule is not associated with a chart of accounts.
  - Can be assigned to an account rule, if the mapping set can have any chart of accounts or no chart of accounts.
  - Cannot be assigned to an account rule, if the account rule is associated to a chart of accounts because the mapping set must have the same chart of accounts.

## Scenario

In the following example, the chart of accounts is set up with four segments. A mapping set is defined with a value set for Supplier Type as described in the following table.

Input Value	Output Value
Services	01-100-6120-000
Consulting	01-400-6110-000

Assume that two invoices are entered into Oracle Fusion Payables, one for a supplier with a type of Services and one for a supplier with a type of Manufacturing. When using the mapping set, the source value Supplier Type from the accounting event data is compared with the mapping set input values to determine the account. In this example, there is a match for the first case; the invoice with a supplier type of Services maps to an input value. However, the invoice with a supplier type of Manufacturing does not map to an input value. The accounts are derived and described in the following table.

Invoice	Supplier Type	Output Value
1	Services	01-100-6120-000
2	Manufacturing	No account generated

---

### Note

To ensure that Transaction 2 is accounted for, the account rule to which the mapping set is assigned may have to be modified. If not, a separate rule can be defined to provide for Supplier Type of Manufacturing or defined a default output in the existing mapping set.

---

## Creating Description Rules: Explained

Use descriptions rules to define the elements of a description that appears on the subledger journal entry at the header and/or the line. The definition determines both the content and sequence in which the elements of the description appear. You can assign a condition to a description rule to determine that the description is selected for display if the condition is satisfied.

### Description Rule Definition

A description rule can be defined with combinations of source and literal values. If sources are used in the rule, the accounting event class associated with the sources determines in which subledger journal entry rule set the description rule can be selected and used.

Build descriptions using the available sources for the application.

The following is the description details that have been entered, using a literal and a source:

- Loan Origination Date = Origination Date
  - Literal = Loan Origination Date
  - Source = Origination Date

Assuming that the source value of the Origination Date is 11/01/11, then a journal entry that has the above description rule attached will have the description, Loan Origination Date 11/01/11.

## Creating Conditions: Examples

The following illustrates an example of defining an account rule with a condition.

### Example 1: Custom Real Estate Application Account Rule Condition Example

This is an example to define an account rule for assignment for a loan journal line. The account rule has two priorities, a mapping set and a constant.

- The first priority will create an output for an account based on the mapping set rule definition.
- A condition is created on the first priority rule. This rule will only be used if the condition below is met.
  - The condition is **Credit Status** must not be null.
  - The accounts derived from the mapping set rule will be used if the Credit Status has a valid value. Otherwise, the accounts derived from the entered constants value from the second priority will be used.

The following table describes the setup of the condition on the first priority:

(	Source	Operator	Value	)
(	"Credit Status"	is not null		)

The second priority will create an output from a constant value (0.9100030.50034206331.0.0.0). There is no condition associated with the second priority.

### Example 2: Oracle Fusion Assets Account Rule Condition Example

This is an example of a rule for a capital purchase. The rule is to be applied only if the distribution account cost center is the same as the liability account cost center and the asset tracking option is Yes. This condition can be expressed as:

- Where Distribution Cost Center = Liability Cost Center and Asset Tracking option = Yes

The following tables describe the setup of the condition:

(	Source	De-limiter	Segment	Operator	Value	De-limiter	Segment	)	And Or
(	"Dis-tribution Account"	.	"Cost Center"	=	"Liability Account"	.	"Cost Center"	)	'AND'
(	"Asset Flag"			=	Yes			)	

The following two rows of data are used in the accounting event, to which the account rule and condition applies.

Account Rule Condition Example: Accounting Event Data

Account	Invoice 1	Invoice 2	Asset Flag
Distribution Account	02-640-2210-1234	01-780-6120-0000	Yes
Liability Account	01-640-2210-0000	02-782-2210-0000	Yes

In the Accounting Event Data table above, assume the cost center segment is the second segment. When the account rule with this condition is used to derive the account for the transaction, the account rule is applied to derive the account of Invoice 1 only. For Invoice 2, even though the assets tracking option is set to Yes, the cost center for the Distribution account and Liability account are not the same. Both conditions must be met in order for the rule to apply.

---

#### Note

When an account source is selected or entered, you must also select or enter a specific segment. If an entire account is required to be used in the condition instead of a specific segment, then select or enter All as the segment for the account.

The condition uses the account source, Distribution Account, and a segment must be provided. In this example, the Cost Center segment is provided.

---

## Creating Supporting References: Explained

Supporting references can be used to store additional source information about a subledger journal entry either at the header or line level.

Sources are assigned to supporting reference segments to indicate which transaction values should be captured on subledger journal entries. The segments are grouped into one supporting reference.

Supporting references that have the option for maintain balances set to **Yes**, establish subledger balances for a particular source and account.

You may want to use Supporting Reference balances for supporting:

- Reconciliation back to the source systems
- Profit and loss balances by dimensions not captured in the chart of accounts

## Supporting Reference Assignment

If the information requirement is purely informational, and not needed for reconciliation or balances, you may consider using description rules to store the source values.

There are several key points to consider when assigning supporting references:

- Define a maximum of five segments for a supporting reference. Assign different sources to each segment.
- Assign only one source from the same accounting event class and application to a supporting reference segment.
- Assign only supporting references with header level sources to the header level of a journal entry rule set.
- Assign supporting references with header and line level sources to the line level of a journal entry rule set.
- Select the balances option in the definition of the supporting reference, to have balances only maintained when the supporting reference is assigned at the line level. For supporting references for which balances are maintained, you can specify whether the balances at the end of a fiscal year are carried forward to the next fiscal year.

As an example:

- A loan information supporting reference can be defined to track two segments:
  - Credit status
  - Loan contract number

Sources will be assigned to each of these segments and the source values for each of these segments will be used to create separate balances.

## Migrate the Configuration

### Migrating Accounting Rules : Points to Consider

Use the export and import functionality in the Setup and Maintenance work area to perform migration of setup data. When migrating accounting rules, you must migrate task lists in entirety and fulfill some requirements.

#### Full Task List Migration

This table shows the task lists to migrate in full, depending on the offering.

Offering	Task List
Oracle Fusion Accounting Hub	Define Accounting Transformation Configuration

Oracle Fusion Financials	Define Subledger Accounting Rules Define Transaction Account Rules
--------------------------	---

There is no support for a partial task list migration.

You may migrate setup data for specific applications only. Note that supporting references, accounting options, and accounting class usages are migrated for all applications, regardless of the applications specified.

### Migration Requirements

Prior to migration, journal entry rule sets and accounting methods must be successfully activated. Invalid journal entry rule sets or accounting methods will cause import failure.

Ensure that your setup data migration includes all dependent business objects from other required setup modules, such as Define Ledgers. The import sequencing of these dependent business objects must be prior to accounting rules business objects.

## Secure Accounting Transformations

### Security for Accounting Transformations: Explained

Accounting transformations require both function and data security privileges.

Oracle Fusion Accounting Hub security for accounting transformations include:

- Setup task security
  - Security to integrate your external systems with accounting transformations, indicating what types of transactions or activities require accounting from those systems.
  - Security to configure accounting rules to define accounting treatments for transactions.
- Transactional task security
  - Security to create subledger journal entries (manual subledger journal entries or those generated by the Create Accounting process).
  - Security to review and generate reports of subledger journal entry headers and lines.

#### Security to Perform Setup Tasks

Use the Define Accounting Transformation Configuration task in the Setup and Maintenance work area to integrate your external systems with the Accounting Hub.

To register your external systems and configure accounting rules, the setup user needs to be provisioned with a role that includes the Accounting Hub Administration Duty role.

- In the security reference implementation, the Financial Application Administrator job role hierarchy includes the Accounting Hub Administration Duty role, which provides the access to integrate your external systems with accounting transformations.
- For more information on available setup job roles, duty roles and privileges, see the Oracle Fusion Accounting Hub Security Reference Manual.

### **Security to Perform Transactional Tasks**

To create and view subledger journal entries, you must have the access necessary to perform the tasks. These tasks can be accessed from the Oracle Fusion General Ledger, Journals work area, therefore you must have access to the work area, and the ledgers (primary, secondary and reporting currency) in which the journal entry is posted.

The following are defined in the security reference implementation:

- The General Accounting Manager job role hierarchy includes duty roles that provide entitlement to manage your general accounting functions. This entitlement provides access to General Ledger Journals work area.
- The General Accounting Manager data role hierarchy includes data security policies that provide entitlement to access ledger and subledger journal entries.
  - Ledger access is provided through Data Access Sets.

The following duty roles need to be assigned directly to the General Accounting Manager job role to provide access to create and view subledger journal entries:

- Subledger Accounting Duty
- Subledger Accounting Reporting Duty

Alternatively, you can assign the Subledger Accounting Duty and Subledger Accounting Reporting Duty roles to any of the following General Ledger job roles:

- Chief Financial Officer
- Controller
- Financial Analyst
- General Accountant

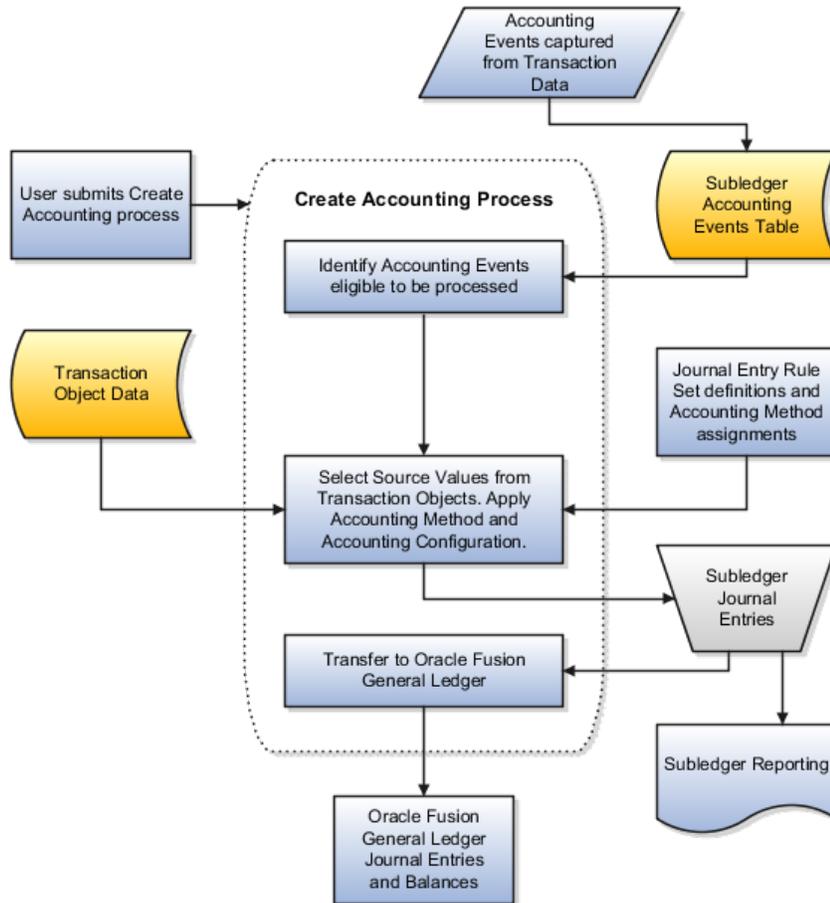
## **Create and Process Subledger Journal Entries**

### **Subledger Journal Entries: How They Are Created and Processed**

The Create Accounting process uses the transaction objects data to create subledger journal entries. For example, if a subledger journal entry rule set

specifies that the customer name should appear in the description of a subledger journal entry line, then the customer name value is taken from the customer name source data provided by the transaction objects.

The following figure illustrates the process used to create subledger journal entries.



## How Subledger Journal Entries Are Created and Processed

1. When transactions are committed in a subledger, accounting events are captured and stored in the subledger accounting events table.
2. The Create Accounting process identifies all accounting events eligible to be processed. For each of these events, the transaction objects provide the Create Accounting process with the transaction objects data (source information). This is the contextual data of the transaction, such as amounts and accounting dates.
3. When the Create Accounting process is run, subledger journal entry rule set definitions and transaction objects data are applied to the transaction object data to create subledger journal entries.
4. These entries are summarized and transferred to Oracle Fusion General Ledger.



# Manage Subledger Accounting

## Manage Subledger Accounting: Overview

In the Manage Subledger Accounting activity, you can generate journal entries for source system transactions, create adjustment entries, and review accounting results.

You can:

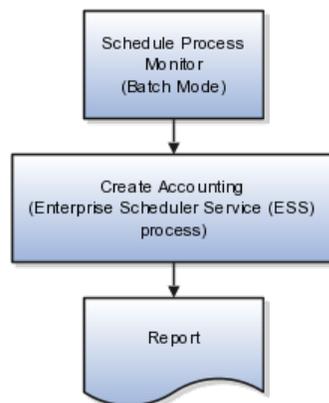
- Create accounting for a batch of transactions by submitting an offline process.
- Create manual adjustment entries.
- Review generated journal entries and projected balances on views and reports.

## Create Accounting

### Submitting the Create Accounting Process: Explained

The Create Accounting process is an Enterprise Scheduler Service (ESS) process. It can be submitted as a request from the Scheduled Processes Monitor window to create journal entries for a batch of events. It has input parameters that determine the execution mode and the selection criteria for events.

The figure below shows the submission of the Create Accounting process.



The following table describes the parameters for the Create Accounting process as submitted in the Scheduled Processes Monitor window.

Prompt	Description
Subledger Application	Source system for which the Create Accounting process is being executed.

Ledger	Ledger name for which the Create Accounting process is being executed.
Process Category	Selecting a process category indicates that all associated accounting event classes and their assigned accounting event types are selected for processing.
End Date	End date puts a filter on the selection of events. Only events having event date before the end date are selected for accounting.  Defaults to SYSDATE.
Accounting Mode	Accounting mode; Draft or Final  Default value is Final.
Process Events	Adds other filter criteria for the Create Accounting process to select events.  All: Process all events.  Errors: Process only those events that have previously been processed in error.  Invalid Accounts: Process only those events that have previously been processed in error. Replace any invalid accounts with the suspense account.  Default value is All.
Report Style	Users can choose to decide on the details of the report. The report can be printed in Summary,, Detail, or No report.  Default value is Summary.
Transfer to General Ledger	Indicates whether the Create Accounting process should submit the Transfer to GL process; Yes, No.  Default value is Yes.
Post in General Ledger	Indicates if users, who have been granted the posting privilege, want to submit General Ledger posting; Yes or No.  Default value is No.
Journal Batch	Batch name used by Transfer to GL to decide on the batch name for the batch created in Oracle Fusion General Ledger.  When a value for the batch name is not provided, journal import defaults will be used.  This is a free text field.
Include User Transaction Identifiers	Default value is No.

## Accessing the Create Accounting Execution Report: Explained

When you submit the Create Accounting process, the Create Accounting Execution Report is submitted automatically upon the completion of the Create

Accounting process in success or in warning status. The Create Accounting process output will contain a message with the Create Accounting Execution Report request identifier. Use this request identifier to access the execution report output.

### **Create Accounting Execution Report**

This report can be recreated as needed by running the process, Create Accounting Report, using the request identifier of the desired Create Accounting process run previously.

If you choose to transfer the entries to Oracle Fusion General Ledger when submitting the Create Accounting process, the report indicates if accounting entries have been transferred or not transferred.

## **Transfer Journal Entries to the General Ledger: Explained**

Importing information from Oracle Fusion subledgers is done using Oracle Fusion Subledger Accounting. Posting from the subledger systems transfers data to the general ledger interface and journal entry tables.

As part of your configuration, you can specify whether the Create Accounting process is to split the creation process into multiple workers (processors). The benefit of splitting the creation process is that accounting can be generated more efficiently with more resources allocated to complete the process. You can have multiple processors running in parallel to create the entries in a shorter period of time.

One restriction is the capacity of the hardware that is being used by the application. The more available processors, the more you are able to allocate to the Create Accounting process.

The decision for how many processors to use is made based upon expected volumes and the processing window, in other words, how much time is allocated to creating accounting. Accounting is often done as a nightly batch process, which has to finish by a certain time so that the systems can be available during work hours on the following day.

There are dependencies between the overall completion status of the Create Accounting process and the workers. In general, the parent does not update to the Completed status until all the workers successfully complete.

The process that allows transfer of subledger journal entries to general ledger uses separate processing workers that are specialized in general ledger transfer. A lightweight general ledger transfer parent process is used to distribute the workload to the workers. In order to transfer entries even faster, you can choose to have a number of parallel processing workers used for high volume general ledger transfer flow. If not specified, the corresponding accounting processors are used.

On the **Manage Subledger Accounting Options** page accessed from the **Manage Subledger Accounting Options** task from your implementation project or the **Setup and Maintenance** work area, select your ledger, click the pencil under **System Options** for Subledger Accounting to set the **Processor Defaults** for the Transfer to GL process.

Column Name	Description
Processing Unit Size	Approximate batch size for high volume general ledger transfer flow. This will also be the general ledger import batch size. If not specified, the corresponding accounting processing unit size will be used.
Number of Processors	Number of parallel processing workers used for high volume general ledger transfer flow. If not specified, the corresponding accounting processes will be used.

---

#### Note

In the **Manage Subledger Accounting Options** page, you can select your subledgers for each ledger or ledger set and specify the **Processing Unit Size** for each event class.

---

## Diagnose Subledger Accounting Event Data: Explained

The diagnostic framework stores data that is used in the creation of a journal entry so that the data can be reported to analyze accounting issues. The purpose of the process is to provide the transaction data that is referenced during accounting through accounting rules and ledger setup.

The diagnostic framework provides a tool that can be used to determine why the accounting was created in a particular way or why there are errors. Errors may occur because either the source system information or the accounting rules are not as expected.

The following examples describe typical uses of the diagnostic framework features.

- In the implementation phase, you can launch the Accounting Event Diagnostic report to review the source values available in the transaction objects.
- On a daily basis, you can use the Accounting Event Diagnostic report to troubleshoot exceptions.

### Diagnostic Framework Features

The diagnostic framework features are as follows:

- **SLA: Diagnostic Enabled:** This option controls whether diagnostic information is gathered by the Create Accounting process. Use with caution. Selecting this option can cause slow performance.
- **Diagnostic Framework Execution:** When the SLA: Diagnostic Enabled option is set to Yes, the diagnostic framework is executed simultaneously with the Create Accounting process. The diagnostic framework data is stored in the diagnostic tables.
- **Accounting Event Diagnostic Report:** To view the diagnostic framework report, users submit the Accounting Event Diagnostic process with the appropriate report parameters.

- Purge Accounting Event Diagnostic Data: Purging is useful when accounting is successfully created so that the volume of information in the report does not increase to a point where it becomes unusable.

Diagnostic framework data purged:

- When the data collected can be purged by running the process.
- When the administrator launches the Purge Accounting Event Diagnostic Data process.

### **Diagnostic Framework Business Process Flow**

The following steps describe the diagnostic framework business process flow:

1. The administrator sets the SLA: Diagnostics Enabled option to Yes for the user or responsibility.
2. Users submit the Create Accounting process that automatically executes the diagnostic framework to populate the diagnostic tables. The diagnostic framework gathers source values and other information from the transaction objects.
3. Users submit the Accounting Event Diagnostic process to view the diagnostic report. Based on the information in the diagnostic report, users may determine if additional or resubmission of information from source systems is required. They may also determine if any updates to the accounting rules is required.
4. Users run the Create Accounting process again to generate subledger journal entries for events that had an error.
5. The administrator submits the Purge Accounting Event Diagnostic Data process to purge the results of the diagnostic framework.

## **Create Subledger Journal Adjustment**

### **Subledger Journal Adjustments: Points to Consider**

Subledger journal entries decrease audit and reconciliation costs, because you can store transaction information directly with journal entries. Using the subledger journal entry validations for online transactions ensures that your user provides the required information in the subledger journal entry header and subledger journal entry lines. The subledger journal entries, combined with accounting rules, offer full disclosure of how the accounting was created for each subledger transaction, and provide assurance that the accounting was done correctly.

You can perform the following actions on your subledger journal entries, depending on their status:

- Edit
- Duplicate
- Reverse

- Delete
- Complete

### **Editing a Subledger Journal Entry**

When editing a subledger journal entry, you can perform the following tasks:

- Edit the journal header information, if the status is not Final.
- Enter new journal lines, including accounts.
- Enter the debit, and credit amounts.
- Enter the accounting class.
- View the impact on general ledger account balances should the entry be completed.
- Post the journal.

You can also edit incomplete manual subledger journal entries.

- Examples of header information which can be updated:
  - Ledger
  - Accounting date
  - Category
  - Description
- Examples of journal entry line information which can be updated:
  - Account
  - Accounting class
  - Entered amount
  - Journal entry lines
- Edit default currency options to be assigned to a manual subledger journal entry.
- Select supporting references and assign values to them.
- Edit or redefine the subledger journal entry description.

### **Duplicating Subledger Journal Entries**

As a time saving feature, you may copy an existing entry.

The duplication feature is enabled for all existing subledger journal entries, regardless of status, and includes the ability to copy complete information required for a manual subledger journal entry header and line.

---

**Note**

All fields can be edited when a manual journal is duplicated.

---

### **Reversing Subledger Journal Entries**

You may reverse manual subledger journal entries in Final status. Reversal options are defaulted from accounting options.

- Switch debit and credit
- Change sign

### **Deleting Subledger Journal Entries**

Oracle Fusion Subledger Accounting provides the ability to delete a manual subledger journal entry that is not in Final status. The ability to delete manual journal entries ensures that users have the flexibility to correct errors without technical support.

### **Completing Subledger Journal Entries**

You can complete manual subledger journal entries in Final and Post to General Ledger status.

## **Creating a Manual Subledger Journal: Points to Consider**

The application enables the user to create manual subledger journal entries online.

### **Creating a Manual Subledger Journal Entry**

This includes the ability to:

- Enter the complete information required for a manual subledger journal entry.
- Enter subledger journal entry descriptions.
- Select a supporting reference and supply the supporting reference value to a subledger journal entry or subledger journal entry line.
- Assign a descriptive flexfield to a subledger journal entry or subledger journal entry line.
- Populate default values for an entered currency for new subledger journal entry line.
- Enter default conversion type, date, and rate information to establish a default currency for the journal that is different than its associated ledger currency.
- View projected balances of entered and accounted journal line amounts.
- Complete and post subledger journal entries.

---

#### **Note**

The ability to post subledger journals to General Ledger is dependent on your security profile. If you do not have the privilege to post, creating a manual subledger journal entry with a Final completion status includes the transfer to General Ledger.

---

## Supporting Reference Assignments: Points to Consider

You may want to analyze account balances and financial results by different transaction attributes. However, transaction information such as sales representative, customer, or vendor are typically not stored in the Oracle Fusion General Ledger because of the volume of general ledger balances it would create, so you are not able to analyze general ledger data categorized by transaction attributes.

You can perform this type of reporting and analysis using supporting reference information from subledger transactions. This feature enables you to create balances based upon transaction attributes not typically stored as segments in the general ledger chart of accounts. For example, you can report on receivables balances by sales representative, customer, credit risk category, item, or any other combination of transaction attributes.

Supporting references can be used to:

- Provide additional information about a subledger journal entry at the header or line level.
- Establish a subledger balance for a particular supporting reference segment value.
- Assist with reconciliation of account balances.
- Provide for financial managerial analysis.

You can assign supporting references at either the subledger journal entry header or the subledger journal entry line.

### Assigning Supporting References at the Subledger Journal Entry Header

Assigning supporting references at the subledger journal entry header level does not keep any balances.

### Assigning Supporting References at the Subledger Journal Entry Line

Assigning supporting references to subledger journal entry lines enables you to maintain detailed balances, by supporting reference value, for general ledger accounts.

## Validating a Third Party Control Account: Examples

If third party control accounts are enabled for the application, and the account entered is a third party control account, you must enter third party information in the journal entry.

## Scenario

For example, if an account is defined as a third party control account with a type of Supplier, then the subledger journal entry lines which use that account must include supplier information. When a valid third party control account is assigned to a journal line, you are required to provide third party information, such as name and site.

Submit the Third Party Balances Report to display subledger balance and account activity information for suppliers and customers. The Customer or Supplier subledger third party balances will be updated when the journal is completed to a Final status.

## Review Subledger Journal Entry

### Subledger Journal Entry: Overview

You can create subledger journal entries by using one of two methods:

- Use the Create Accounting process to process accounting events using accounting rules.
- Create manual subledger journal entries.

Subledger journal entries are always created in a given subledger application context. When the subledger journal entry is complete, the appropriate sequence names and numbers are assigned, and the corresponding secondary ledger and reporting currency journal entries are created if applicable.

Manual journal entries can be entered for primary ledgers or for secondary ledgers. Manual journals for primary ledgers are automatically created only for associated reporting currencies, not secondary ledgers.

### Reviewing a Subledger Journal Entry: Points to Consider

You have the ability to review subledger journal entries, whether they were created from processing accounting events, or manually entered. You may query subledger journal entries directly, or locate them via searches for journal entries with a specific status, unprocessed accounting events, or subledger journal entry lines. Advanced search functionality, including the ability to use multiple search criteria is available.

#### Review Subledger Journal Entries

Perform an inquiry on unprocessed accounting events, subledger journal entries and subledger journal entry lines based on multiple selection criteria.

- Create, edit, duplicate, reverse or delete a manual subledger journal entry
- View detailed information about the subledger journal entry
- View a subledger journal entry in the T-Accounts format

- View transactions underlying the accounting event or the subledger journal entry
- View supporting references associated with the subledger journal entry and lines
- View tagged subledger journal entries or create a tag on the subledger journal entry

## Viewing Projected Balances: Points to Consider

Use the projected balances feature to view the impact on account balances for selected subledger journal entry lines.

The projected balances flow has the following business benefits:

- Creation and validation of unposted manual journal entries by providing knowledge users with immediate and relevant information about the account balances for the selected journal lines.
- Validation and reconciliation of posted journal entries by providing immediate and relevant information about the account balances for the selected journal lines.

### Projected Balances

Oracle Fusion Subledger Accounting manual journal entry and Oracle Fusion General Ledger manual journal entry, approval, and inquiry pages display projected or current balances including the current journal entry line amounts. Depending on whether the journal is posted or not, the current balance (for the period of the journal) is displayed or calculated.

The projected balance region in the contextual area is display the projected balances for the account that includes the amounts of the selected journal entry line. Additionally, if more than one journal line for same account of the selected journal line exists, then the projected balance amount will include the impact from each journal line. The Period To Date, Quarter To Date, and Year To Date balances are also available.

- For unposted journals, the period balance is projected by summing the current balance with the subledger journal entry line amounts
- For posted journals, the opening balance and the period activity is calculated using the current balance and journal line amount

Projected balances increases accuracy when booking entries to reconcile accounts.

## Subledger Accounting Reports: Overview

Oracle Fusion Subledger Accounting provides accounting reports for fiscal and internal control purposes.

The subledger accounting reports include subledger journals for the journal sources that use Oracle Fusion Subledger Accounting and Oracle Fusion General

Ledger journals for journal sources such as allocations and revaluation that do not originate within General Ledger.

The reports are comprehensive from a financial standpoint and include the best source of information for each type of journal entry. These reports can therefore be used in lieu of the General Ledger based journals and account analysis reports if there is a need to see detailed subledger journal entries as well as supporting transaction information.

The following reports are available for Oracle Fusion Subledger Accounting:

Title	Process Name	Description
Account Analysis Report	XLAAARPT	Provides drill down information about the movement on a particular account for an accounting period or range of accounting periods. It only includes journal entries posted to Oracle Fusion General Ledger.
Accounting Event Analysis Report	XLAAPDIAG	Shows the transaction data used in accounting. This report is intended to be used when there are accounting errors and the setup and the transaction data needs to be analyzed to diagnose the cause of the errors.  The data is collected when the profile option SLA: Enable Diagnostics is set and the Create Accounting process is run.
Journal Entries Report	XLAJELRPT	Displays detailed information for subledger and general ledger journal entries.
Subledger Period Close Exceptions Report	XLAPEXRPT	Lists all accounting events and journal entries that fail period close validation. Reasons for failure include incomplete transactions, unprocessed accounting events, invalid journal entries, and journal entries that have not been transferred.
Third Party Balances Report	XLATPRPT	Displays subledger balance and account activity information for suppliers and customers.

These reports are classified as Oracle Business Intelligence Publisher (Oracle BI Publisher) and are scheduled and run from the Scheduled Processes work area on the Navigator menu.

1. Click the **Schedule New Process** button.
2. Search on the report **Title** or **Process Name**.
3. Enter the desired parameters.
4. Enter the desired process options and schedule.

5. Click **Submit**.

---

---

# Define and Maintain Intercompany Processing Rules

## Manage Intercompany System Options

### Intercompany System Options: Explained

Define intercompany system options to set up intercompany processing rules at the enterprise level, based on your specific business needs.

In order to maintain consistency throughout an enterprise, intercompany transaction processing rules should be defined at the enterprise level. By standardizing these rules, an enterprise can minimize disputes, decrease processing time, and cut administrative costs.

Before setting up intercompany system options, you need to determine how you will process your intercompany transactions, for example, to:

- Enforce an enterprise-wide currency or allow intercompany transactions in local currencies.
- Allow receivers to reject intercompany transactions.
- Determine the minimum transaction amount that will be processed.

---

#### Note

Important: Changing and saving a system option will have no effect on intercompany transactions already in progress. Intercompany system options will only affect new intercompany transactions, and are not retroactive to any transactions previously entered.

---

The system options are:

- Intercompany Batch Numbering
- Intercompany Currency
- Minimum Transaction Amount

- Conversion Rate Type
- Allow Receiver to Reject Transactions
- Intercompany Calendar and Period Type
- Default Transaction Type

### **Intercompany Batch Numbering**

The intercompany batch numbering option defines whether to use system generated or manual transaction batch numbering.

Select the **System Generated** option to use only one automatic sequence number in intercompany within an instance. The sequence numbers are unique within the instance. All transactions created in intercompany will use this sequence.

Use the **Manual** option to manually enter a batch number, up to 20 characters.

### **Intercompany Currency**

Standardize transaction processing by selecting an intercompany currency. Intercompany transactions are always entered in this currency in all legal entities within the enterprise. This simplifies transaction processing, and eliminates foreign exchange rate fluctuation risks. If an intercompany currency is selected, it will populate and default value and overwrite any existing value in Minimum Transaction Currency.

You may not update the Minimum Transaction Currency if intercompany Currency is entered.

### **Minimum Transaction Amount**

The minimum transaction amount represents a minimum threshold intercompany transaction amount, and prevents the submission of immaterial transactions for small amounts, which are non-value added. In order to implement this rule, you must select a minimum transaction currency for processing intercompany transactions. These two system options must be related to ensure that when comparing a transaction amount to the minimum transaction amount, the two numbers are entered in the same currency, allowing for an accurate comparison.

### **Conversion Rate Type**

Choose a conversion rate type if you enter intercompany transactions in a foreign currency. This rate type is used to convert all foreign currency transactions to functional equivalents when transactions are transferred to general ledger, receivables and payables. To ensure that intercompany accounts remain balanced, set a corporate-wide conversion rate type that cannot be changed by any intercompany trading partner within the enterprise.

### **Allow Receiver to Reject Transactions**

Use this system option to determine if receivers of intercompany transactions can reject transactions or not. For example, if your company policy requires intercompany transactions be approved, but do not allow receivers to reject the transactions, then you can use this system option to implement your policy.

## Intercompany Calendar and Period Type

You can have an intercompany calendar that is separate from the general ledger calendar. This ensures that the opening and closing of periods can be controlled separately from the general ledger calendar. The Period Type value is defaulted from the selected accounting calendar and cannot be updated. It indicates the type of period defined for that calendar.

If an intercompany calendar is chosen, the intercompany transactions will be validated against this calendar for period open and close.

In order to update the intercompany calendar to use a different general ledger accounting calendar, the intercompany period status for all intercompany transaction types must either be Never Opened or Closed.

### Default Transaction Type

Optionally select one of the enabled intercompany transaction types to use as the default type for all new intercompany transactions.

## Manage Intercompany Organizations

### Managing Intercompany Organizations: Points to Consider

The Manage Intercompany Organizations task allows you to define the legal entities that are classified as intercompany organizations. The intercompany organization can act either as a provider or a receiver in an intercompany transaction.

Optionally assign a receivables and payables business unit to the organization if you require invoice generation. When you create a transaction for this organization use an invoicing transaction type. Invoices will be generated in Oracle Fusion Receivables and Oracle Fusion Payables for the business units specified.

You can initiate an intercompany transaction only for those organizations that you have access to. If a new organization is added after the system is configured, the intercompany accountant or system administrator should ensure that access is given to the appropriate users.

You can disable the intercompany organization if there are no open transactions for the organization.

When creating an organization, the following attributes should be considered:

- Legal Entity
- Receivables and Payables Business Units
- Default Organization Contact

### Legal Entity

Each intercompany organization must be associated with a legal entity, but you can associate more than one organization to a legal entity.

## **Receivables and Payables Business Units**

The available business units are those associated with the ledger to which the selected legal entity belongs. This assignment is optional, but is needed when the organization is the provider, and intercompany invoices are required.

If you enter the wrong receivables or payables business units, you can still correct them as long as the organization is not yet used in an intercompany transaction, regardless of the transaction status.

The organization can be disabled if there are no open transactions, and all transactions for that organization are having either New or Complete status.

### **Default Organization Contact**

The default organization contact is the contact person assigned to the intercompany organization. The contact person can be assigned to one or more organizations.

## **Define Invoicing Options**

## **Maintaining Customer and Supplier Assignments: Points to Consider**

The intercompany customer and supplier assignments are used to identify each legal entity and the customer and supplier each legal entity represents. The assignments are used to derive the customers and suppliers for intercompany invoicing.

When intercompany invoicing is required for the intercompany transaction type, you must associate a customer and a supplier with the legal entities of the provider and receiver of the intercompany transaction, so that receivables and payables invoices can be generated. You can associate a legal entity with either a customer account or supplier, or both.

### **Customer Account**

Assign a unique customer account to the legal entity of the organization that receives and approves intercompany transactions. The customer must have an active site, and it must not be an external customer.

### **Supplier**

Assign a unique supplier to the legal entity of the organization that initiates intercompany transactions. The supplier must have an active primary pay site.

### **Additional Considerations**

A customer or a supplier can be associated with only one legal entity. Once a customer or a supplier is associated with a legal entity, it cannot be associated with another legal entity.

The customer account and supplier assigned to the legal entity can be modified at any time, regardless of the existing transaction status.

## Intercompany Receivables Assignments: Explained

Use Oracle Fusion Intercompany to generate invoices for intercompany transactions.

### Defining Receivables Assignments

Set up your receivables assignments by mapping an intercompany transaction type and a receivables business unit to the receivables transaction type and receivables memo line. Oracle Fusion Receivables will use the receivables transaction type and the receivables memo line to process intercompany transactions transferred to the receivables application. You can configure specific receivables transaction types, and receivables memo lines, that you use for each intercompany transaction type for a receivables business unit. Invoices can then be transferred to Oracle Fusion Payables and recorded there.

Intercompany provides a default receivables transaction type of Intercompany and a default receivables memo line of Global Intercompany. These defaults are used when there are no other assignments. However, you can choose to set up individual assignments for each receivables business unit and intercompany transaction type to override the default values.

First, select a business unit and intercompany transaction type, and then select the receivables transaction type and receivables memo line. The receivables transaction type values available are derived from the reference data set for the receivables transaction type assigned to the business unit. The receivables memo line values are derived from the reference data set for receivables memo line assigned to the business unit.

## Manage Intercompany Balancing Rules

### Intercompany Balancing Rules: Explained

Intercompany balancing rules are used to generate the accounts needed to balance journals that are out of balance by legal entity or primary balancing segment values.

You specify the intercompany receivables and intercompany payables accounts you want to use. The intercompany balancing feature then uses these rules to generate the accounts of the balancing lines it creates.

### Defining Intercompany Balancing Rules

You can define intercompany balancing rules at the following rule levels:

1. Primary balancing segment
2. Legal entity
3. Ledger
4. Chart of accounts

The rules are evaluated in the order shown above. For example, you can define a Primary Balancing Segment rule and a Legal Entity level rule. If both rules are used to balance a particular journal, the Primary Balancing Segment rule is used, as it has a higher precedence.

You have flexibility in defining your intercompany balancing rules. You can have a simple setup in which you define one rule for your chart of accounts. This rule is used for all intercompany balancing for all ledgers that use this chart of accounts. Alternatively, you can have a more granular set of rules. For example, you can define a different rule for each legal entity and one chart of accounts rule to cover any gaps in your rule definitions. You can gain even more granularity by defining rules for specific journal and/or category combinations or intercompany transaction types.

## Intercompany Balancing Rules: Examples

This topic provides examples of intercompany balancing rules and the intercompany balancing lines generated. These rules are used to generate the accounts needed to balance journals that are out of balance by legal entity or primary balancing segment values.

### Simple Chart of Accounts

In this scenario you have one chart of accounts for all ledgers. The chart of accounts has an intercompany segment. You are using this intercompany segment and the company segment to identify the intercompany trading partners for each transaction. You do not have a need to track their intercompany activity at a granular level such as by journal source and journal category or by intercompany transaction type.

Setup

- InFusion USA Chart of Accounts

Segment Qualifier	Primary Balancing Segment	Second Balancing Segment	Third Balancing Segment	Account	Intercompany Segment
Segment Name	Company (CO)	Cost Center (CC)	Product (PROD)	Account (ACCT)	Intercompany (IC)

- Ledger, Legal Entity, Primary Balancing Segment Value Assignments

Ledger	Legal Entity	Primary Balancing Segment Value
InFusion USA	InFusion Farms	3100, 3200, 3300, 3400, 3500
InFusion USA	InFusion Textiles	4000
InFusion USA	InFusion Products (East)	5000
InFusion USA	InFusion Products (West)	6000
InFusion USA		1000, 9000

- Chart of Accounts Rule

Rule Number	Chart of Accounts	AR Account	AP Account	Source	Category	Transaction Type
1	InFusion USA Chart of Accounts	1000 - 000 - 0000 - 13010 - 0000	1000 - 000 - 0000 - 21010 - 0000	Other	Other	None

- Journal Balancing
- Journal before Balancing

Line	Line Type	Legal Entity	CO	CC	PROD	ACCT	IC	Debit	Credit
1	Expense	InFusion Farms	3100	100	1200	52330	0000	150	
2	Liability	InFusion Textiles	4000	500	1300	40118	0000		150

- Journal Balancing
- Journal after Balancing

Uses Rule	Line	Line Type	Legal Entity	Compan	Cost Center	Product	Account	Intercom	Debit	Credit
	1	Expense	InFusion Farms	3100	100	1200	52330	0000	150	
	2	Liability	InFusion Textiles	4000	500	1300	40118	0000		150
1	3	IC AP	InFusion Farms	3100	100	1200	21010	4000		150
1	4	IC AR	InFusion Textiles	4000	500	1300	13010	3100	150	

### Legal Entity and Chart of Accounts Rules

In this example the legal Entity InFusion Textiles intercompany manufacturing activities are tracked separately from its non-manufacturing activities. In order to achieve this legal entity level rules are defined specifically between the legal entity InFusion Textiles and the two manufacturing legal entities, InFusion Products (East) and InFusion Products (West). A chart of accounts rule is created to cover all other intercompany activities.

Setup

- InFusion USA Chart of Accounts

Segment Qualifier	Primary Balancing Segment	Second Balancing Segment	Third Balancing Segment	Account	Intercompany Segment
Segment Name	Company	Cost Center	Product	Account	Intercompany

- Ledger, Legal Entity, Primary Balancing Segment Value Assignments

Ledger	Legal Entity	Primary Balancing Segment Value
InFusion USA	InFusion Farms	3100, 3200, 3300, 3400, 3500
InFusion USA	InFusion Textiles	4000
InFusion USA	InFusion Products (East)	5000
InFusion USA	InFusion Products (West)	6000
InFusion USA		1000, 9000

- Chart of Accounts Rule

Rule Number	Chart of Accounts	AR Account	AP Account	Source	Category	Transaction Type
2	InFusion USA Chart of Accounts	1000 - 000 - 0000 - 13050 - 0000	1000 - 000 - 0000 - 21050 - 0000	Other	Other	None

- Legal Entity Level Rule

Rule No.	From Legal Entity	To Legal Entity	AR Account	AP Account	Source	Category	Transaction Type
3	InFusion Textiles	InFusion Products (West)	1000 - 000 - 0000 - 13020 - 0000	1000 - 000 - 0000 - 21020 - 0000	Other	Other	None
4	InFusion Textiles	InFusion Products (East)	1000 - 000 - 0000 - 13030 - 0000	1000 - 000 - 0000 - 21030 - 0000	Other	Other	None

- Journal Balancing

- Journal before Balancing

Line	Line Type	Legal Entity	CO	CC	PROD	ACCT	IC	Debit	Credit
1	Expense	InFusion Farms	3100	100	1200	52330	0000	150	
2	Expense	InFusion Products (East)	5000	100	1200	52340	0000	200	
3	Expense	InFusion Products (West)	6000	200	1300	52345	0000	300	
4	Liability	InFusion Textiles	4000	500	1300	40118	0000		650

- Journal Balancing

- Journal after Balancing

Uses Rule	Line	Line Type	Legal Entity	CO	CC	PROD	ACCT	IC	Debit	Credit
	1	Expense	InFusion Farms	3100	100	1200	52330	0000	150	
	2	Expense	InFusion (East)	5000	100	1200	52340	0000	200	
	3	Expense	InFusion (West)	6000	200	1300	52345	0000	300	
	4	Liability	InFusion Textiles	4000	500	1300	40118	0000		650
2	5	IC AR	InFusion Textiles	4000	500	1300	13050	3100	150	
2	6	IC AP	InFusion Farms	3100	100	1200	21050	4000		150
4	7	IC AR	InFusion Textiles	4000	500	1300	13030	5000	200	
2	8	IC AP	InFusion	5000	100	1200	21050	4000		200
3	9	IC AR	InFusion Textiles	4000	500	1300	13020	6000	300	
2	10	IC AP	InFusion (West)	6000	200	1300	21050	4000		300

## Using Chart of Accounts Rules for Intercompany Balancing: Examples

Use chart of accounts rules for intercompany balancing. You have flexibility in defining your intercompany balancing rules with the setup of a single chart of accounts rule to use for all ledgers that use this chart of accounts.

When you create a chart of accounts rule, you specify the chart of accounts, intercompany receivables, and intercompany payables accounts you want to use, as well as the source and category. It is recommended that the intercompany receivables account be an asset type account, and the intercompany payables account be a liability type account.

You can define rules that are applied to a specific source and category, such as Payables and Invoices, or a specific intercompany transaction type, such as Intercompany Sales. Alternatively, you can choose to create rules for all sources and categories by selecting the source of **Other** and the category of **Other**.

You can have a more complex structure and define multiple rules between pairs of ledgers, legal entities, or primary balancing segment values. If you choose to have rules at various levels, then intercompany balancing evaluates the rules in the following order.

1. Primary balancing segment rules
2. Legal entity level rules
3. Ledger level rules
4. Chart of accounts rules

It is therefore recommended that you set up a chart of accounts rule for every chart of accounts structure you have. This will ensure that Intercompany Balancing will always find a rule to use to generate balancing accounts.

Intercompany Balancing will then evaluate the journal source and journal category combination in determining which rule to use for balancing. The order of precedence is as follows.

- Specific journal source and journal category
- Specific journal source and journal category of Other
- Journal source of Other and specific journal category
- Journal source of Other and journal category of Other

### Chart of Accounts Rule Example

In this scenario, you choose to track intercompany balancing for companies with values 3000, and 4000 to separate intercompany accounts. You will set up specific rules are set up at the primary balancing segment value level for this. A chart of accounts rule is created for all other intercompany activity.

Setup

InFusion USA Chart of Accounts

Segment Qualifer	Primary Balancing Segment	Balancing Segment 2	Segment	Segment	Intercompany Segment
Segment Name	Company (CO)	Cost Center (CC)	Product (PROD)	Account (ACCT)	Intercompany (IC)

Ledger, Legal Entity, and Primary Balancing Segment Value Assignments

Ledger	Legal Entity	Primary Balancing Segment Value
InFusion USA	Infusion Farms	3000
InFusion USA	InFusion Textiles	4000
InFusion USA	InFusion Production	5000
InFusion USA		1000, 9000

Chart of Accounts Rule

Rule No. 1

- Chart of Accounts: InFusion USA
- Source: Other
- Category: Other
- Transaction Type: None

IC Account	CO	CC	PROD	ACCT	IC
AR Account	1000	000	0000	13010	0000
AP Account	1000	000	0000	21010	000

Primary Balancing Segment Rules

Rule No. 2

- From Ledger and To Ledger: InFusion USA
- From Primary Segment Value: 3000
- To Primary Segment Value: 4000
- Source: Other
- Category: Other
- Transaction Type: None

IC Account	CO	CC	PROD	ACCT	IC
AR Account	1000	000	0000	13011	0000
AP Account	1000	000	0000	21011	0000

Rule No. 3

- From Ledger and To Ledger: InFusion USA
- From Primary Segment Value: 3000
- To Primary Segment Value: 5000
- Source: Other
- Category: Other
- Transaction Type: None

IC Account	CO	CC	PROD	ACCT	IC
AR Account	1000	000	0000	13012	0000
AP Account	1000	000	0000	21012	0000

Rule No. 4

- From Ledger and To Ledger: InFusion USA
- From Primary Segment Value: 4000
- To Primary Segment Value: 3000
- Source: Other
- Category: Other

- Transaction Type: None

IC Account	CO	CC	PROD	ACCT	IC
AR Account	1000	000	0000	13013	0000
AP Account	1000	000	0000	21013	0000

Rule No. 5

- From Ledger and To Ledger: InFusion USA
- From Primary Segment Value: 4000
- To Primary Segment Value: 5000
- Source: Other
- Category: Other
- Transaction Type: None

IC Account	CO	CC	PROD	ACCT	IC
AR Account	1000	000	0000	13014	0000
AP Account	1000	000	0000	21014	0000

Intercompany Balancing Lines Generated for Out of Balance Journal No. 1

- Source: Manual
- Category: Adjustment

Line	Account	Debit	Credit	Description	Uses Rule No.
1	5000- 100- 1200- 52330- 0000	150			
2	4000- 110- 1200- 41111- 0000		150		
3	5000- 100- 0000- 21010- 4000		150	Intercompany Payables	1
4	4000- 110- 0000- 13014- 5000	150		Intercompany Receivables	5

Intercompany Balancing Lines Generated for Out of Balance Journal No. 2

- Source: Manual
- Category: Adjustment

Line	Account	Debit	Credit	Description	Uses Rule No.
1	3000- 100- 1200- 52330- 0000	150			
2	4000- 110- 1200- 41111- 0000		150		

3	3000- 100- 0000- 21011- 4000		150	Intercompany Payables	2
4	4000- 110- 0000- 13013- 3000	150		Intercompany Receivables	4

## Manage Ledger Balancing Options

### Defining Ledger Balancing Options: Explained

Ledger balancing options are defined for the ledger to balance the second balancing segment and/or the third balancing segment, when a transaction is unbalanced by one of these segments.

Ledger balancing options include the following settings:

- Oracle Fusion Receivables and Oracle Fusion Payables accounts used for ledger balancing
- Summarization options
- Clearing company options

#### Receivables and Payables Accounts used for Ledger Balancing

You can choose to specify the receivables and payables accounts to be used, if your chart of accounts has the second balancing segment and/or the third balancing segment enabled. These accounts are used for the balancing lines generated when a journal is balanced by its primary balancing segment values but is not balanced by its second balancing segment and/or third balancing segment.

#### Summarization Options

You can choose to summarize balancing lines generated for a primary balancing segment out of balance scenario, where all the primary balancing segment values are assigned to the same legal entity, by specifying the Summarization option of Summary Net or Detail. You can choose to summarize by primary balancing segment value or alternatively have individual balancing lines (that have not been summarized) generated. Note that summarization always applies to balancing lines generated in a cross legal entity scenario.

#### Clearing Company Options

You can choose to set clearing company options to balance a journal with different primary balancing segment values that all belong to a single legal entity. Set the following options to handle your clearing company balancing.

- Clearing Company Condition
  - Choose to balance using a clearing company value for all journals or for journals with many legal entities on the debit side and many legal entities on the credit side.

- The default value for this option is to error Many-to-Many journals.
- Clearing Company Source
  - Choose how the clearing company value is derived for your balancing lines, from the following options:
    - Default clearing balancing segment value.
      - Choose this option if you want a single specific primary balancing segment value for your clearing company.
    - Default Rule.
      - Choose this option if you want to allow the system to derive the clearing company value from a default intercompany balancing rule.
    - Manually entered clearing balancing segment value.
      - Choose this option if you want to enter the clearing company value when you create a journal.
- Clearing Company Value
  - If you chose the default clearing balancing segment value as your clearing company source, you can enter your chosen primary balancing segment value in this field.

## Defining Ledger Balancing Options: Examples

This topic provides examples of ledger balancing options, the setup required, and the journal before and after balancing.

### Simple Ledger Balancing with no Clearing Company Options

In this scenario the enterprise has the second balancing segment and the third balancing segment enabled for its chart of accounts. The journal is balanced by primary balancing segment but is out of balance by the second balancing segment and the third balancing segment.

Setup

- InFusion USA Chart of Accounts

Segment Qualifier	Primary Balancing Segment	Second Balancing Segment	Third Balancing Segment	Account	Intercompany Segment
Segment Name	Company (CO)	Cost Center (CC)	Product (PROD)	Account (ACCT)	Intercompany (IC)

- Ledger, Legal Entity, Primary Balancing Segment Value Assignments

Ledger	Legal Entity	Primary Balancing Segment Value
InFusion USA	InFusion Farms	3100, 3200, 3300, 3400, 3500
InFusion USA	InFusion Textiles	4000
InFusion USA	InFusion Products (East)	5000
InFusion USA	InFusion Products (West)	6000
InFusion USA		1000, 9000

- Ledger Balancing Options

Rule Number	Ledger	Source	Category	Transaction Type	AR Account	AP Account
1	InFusion USA	Other	Other	None	1000 - 000 - 0000 - 13010 - 0000	1000 - 000 - 0000 - 21010 - 0000

- Journal Balancing
  - Journal Before Balancing

Line	Line Type	Legal Entity	CO	CC	PROD	ACCT	IC	Debit	Credit
1	Expense	InFusion Farms	3100	100	1200	52330	0000	150	
2	Liability	InFusion Farms	3100	500	1300	40118	0000		150

- Journal Balancing
  - Journal after Balancing

Uses Rule	Line	Line Type	Legal Entity	CO	CC	PROD	ACCT	IC	Debit	Credit
	1	Expense	InFusion Farms	3100	100	1200	52330	0000	150	
	2	Liability	InFusion Farms	3100	500	1300	40118	0000		150
1	3	AP	InFusion Farms	3100	100	1200	21010	0000		150
1	4	AR	InFusion Farms	3100	500	1300	13010	0000	150	

### Ledger Balancing Options with Detail Summarization and Clearing Company Options Set

In this scenario the enterprise has the second balancing segment and the third balancing segment enabled for its chart of accounts. Management has decided to use a clearing company for balancing Many-to-Many journals only. Since the primary balancing segment values in the journal are out of balance

intercompany balancing is required. Additionally, since clearing company options have been specified they will be used to balance the journal. Note that if the primary balancing segment values were balanced and only the second balancing segment and the third balancing segment were out of balance, the clearing company options would not be used.

Setup

- InFusion 1000, USA Chart of Accounts

Segment Qualifier	Primary Balancing Segment	Second Balancing Segment	Third Balancing Segment		Intercompany Segment
Segment Name	Company	Cost Center	Product	Account	Intercompany

- Ledger, Legal Entity, Primary Balancing Segment Value Assignments

Ledger	Legal Entity	Primary Balancing Segment Value
InFusion USA	InFusion Farms	3100, 3200, 3300, 3400, 3500
InFusion USA	InFusion Textiles	4000
InFusion USA	InFusion Products (East)	5000
InFusion USA	InFusion Products (West)	6000
InFusion USA		1000, 9000

- Chart of Accounts Rule

Rule Number	Chart of Accounts	AR Account	AP Account	Source	Category	Transaction Type
1	InFusion USA Chart of Accounts	1000 - 000 - 0000 - 13050 - 0000	1000 - 000 - 0000 - 21050 - 0000	Other	Other	None

- Ledger Balancing Options

Rule Number	Ledger	Source	Category	Transaction Type	AR Account	AP Account
2	InFusion USA	Other	Other	None	1000 - 000 - 0000 - 13010 - 0000	1000 - 000 - 0000 - 21010 - 0000

- Clearing Company Options

Rule Number	Ledger	Source	Category	Transaction Type	Condition	Source	Value
2	InFusion USA	Other	Other	None	Use for many-to-many journals only	Default clearing balancing segment value	9000

---

**Note**

The Ledger Balancing Options and Clearing Company Options appear as one line on the page.

---

- Journal Balancing
  - Journal before Balancing

Line	Line Type	Legal Entity	CO	CC	PROD	ACCT	IC	Debit	Credit
1	Expense	InFusion Farms	3100	100	1200	52330	0000	150	
2	Expense	InFusion Farms	3100	300	1200	52340	0000	200	
3	Expense	InFusion Farms	3300	200	1300	52345	0000	300	
4	Liability	InFusion Farms	3400	500	1300	40118	0000		320
5	Liability	InFusion Farms	3500	600	1400	40112	0000		330

- Journal Balancing
  - Journal after Balancing

Uses Rule	Line	Line Type	Legal Entity	CO	CC	PROD	ACCT	IC	Debit	Credit
	1	Expense	InFusion Farms	3100	100	1200	52330	0000	150	
	2	Expense	InFusion Farms	3100	300	1200	52340	0000	200	
	3	Expense	InFusion Farms	3300	200	1300	52345	0000	300	
	4	Liability	InFusion Farms	3400	500	1300	40118	0000		320
	5	Liability	InFusion Farms	3500	600	1400	40112	0000		330
1	6	IC AR		9000	000	0000	13050	3100	150	
1	7	IC AP		3100	100	1200	21050	9000		150
1	8	IC AR		9000	000	0000	13050	3100	200	
1	9	IC AP		3100	300	1200	21050	9000		200
1	10	IC AR		9000	000	0000	13050	3300	300	
1	11	IC AP		3300	200	1300	21050	9000		300
1	12	IC AR		3400	200	1300	13050	9000	320	
1	13	IC AP		9000	000	000	21050	3400		320
1	14	IC AR		3500	600	1400	13050	9000	330	
1	15	IC AP		9000	000	000	21050	3500		330

# Manage Intercompany Transactions

## Generating Intercompany Receivables and Intercompany Payables Accounts for Manual Transactions: Examples

The receivables (AR) and payables (AP) accounts for manual intercompany transactions are generated automatically by Oracle Fusion Intercompany. Enter distributions for the transaction and intercompany generates the receivables and payables accounts, based on the intercompany balancing rules setup.

Intercompany uses the attributes of the batch, such as transaction type, provider and receiver legal entities, to ascertain which rule to use. Intercompany then uses the rule, and the segment details of the first distribution account for the provider, to build the intercompany account combination for the provider side of the transaction. Similarly, intercompany builds the intercompany account for the receiver side of the transaction, based on the first receiver distribution account.

Intercompany will evaluate the rules in the following order.

1. Primary balancing segment rules
2. Legal entity level rules
3. Ledger level rules
4. Chart of accounts rules

If there is no matching rule at the lower levels, then intercompany will use the chart of accounts rule. It is therefore recommended that you set up a chart of accounts rule for every chart of accounts structure you have. This will ensure that intercompany will always find a rule to use to generate the intercompany receivables and intercompany payables accounts for transactions.

Intercompany will then evaluate the transaction type in determining which rule to use to generate the receivables or payables account. A rule with a specific transaction type takes precedence over a rule defined for the All Other transaction type.

### Generating Intercompany Receivables and Intercompany Payables Accounts for Manual Transactions Example

In this scenario you choose to track your intercompany sales for the farming and textile companies separately from other intercompany activities. Separate intercompany accounts are used for these two companies. A chart of accounts rule is created for all other intercompany activity.

Setup

InFusion USA Chart of Accounts

Segment Qualifier	Primary Balancing Segment	Balancing Segment 2	Segment	Segment	Intercompany Segment
Segment Name	Company (CO)	Cost Center (CC)	Product (PROD)	Account (ACCT)	Intercompany (IC)

Ledger, Legal Entity, and Primary Balancing Segment Value Assignments

Ledger	Legal Entity	Primary Balancing Segment Value
InFusion USA	Infusion Farms	3100, 3200, 3300, 3400, 3500
InFusion USA	InFusion Textiles	4000
InFusion USA	InFusion Production	5000
InFusion USA		1000, 9000

Chart of Accounts Rule

Rule No. 1

- Chart of Accounts: InFusion USA
- Source: None
- Category: None
- Transaction Type: All Other

IC Account	CO	CC	PROD	ACCT	IC
AR Account	1000	000	0000	13020	0000
AP Account	1000	000	000	21020	000

Legal Entity Rules

Rule No. 2

- From Ledger and To Ledger: InFusion USA
- From Legal Entity: InFusion Farms
- To Legal Entity: InFusion Textiles
- Source: None
- Category: None
- Transaction Type: Intercompany (IC) Sales

IC Account	CO	CC	PROD	ACCT	IC
AR Account	1000	000	0000	13011	0000
AP Account	1000	000	0000	21011	0000

Rule No. 3

- From Ledger and To Ledger: InFusion USA
- From Legal Entity: InFusion Farms
- To Legal Entity: InFusion Production
- Source: None

- Category: None
- Transaction Type: Intercompany Sales

IC Account	CO	CC	PROD	ACCT	IC
AR Account	1000	000	0000	13012	0000
AP Account	1000	000	0000	21012	0000

Rule No. 4

- From Ledger and To Ledger: InFusion USA
- From Legal Entity: InFusion Textiles
- To Legal Entity: InFusion Farms
- Source: None
- Category: None
- Transaction Type: Intercompany Sales

IC Account	CO	CC	PROD	ACCT	IC
AR Account	1000	000	0000	13013	0000
AP Account	1000	000	0000	21013	0000

Rule No. 5

- From Ledger and To Ledger: InFusion USA
- From Legal Entity: InFusion Textiles
- To Legal Entity: InFusion Production
- Source: None
- Category: None
- Transaction Type: Intercompany Sales

IC Account	CO	CC	PROD	ACCT	IC
AR Account	1000	000	0000	13014	0000
AP Account	1000	000	0000	21014	0000

Intercompany Accounts Generated for Intercompany Debit Transactions

Transaction Type	Provider LE	Receiver LE	Provider Distribution	Provider AR Account Generated	Uses Rule No.
IC Sales	InFusion Farms	InFusion Textiles	3100- 100- 1200- 52330- 0000	3100- 100- 0000- 13011- 4000	2
IC Adjustments	InFusion Farms	InFusion Textiles	3100- 100- 1200- 52330- 0000	3100- 100- 0000- 13020- 4000	1

IC Sales	InFusion Production	InFusion Farms	5000- 120- 1300- 52345- 0000	5000- 120- 0000- 13020- 3200	1
----------	---------------------	----------------	------------------------------	------------------------------	---

This table displays the Receiver side of the transaction.

Transaction Type	Provider LE	Receiver LE	Receiver Distribution	Receiver AR Account Generated	Uses
IC Sales	InFusion Farms	InFusion Textiles	4000- 110- 1200- 41111- 0000	4000- 110- 0000- 21013- 3100	4
IC Adjustments	InFusion Farms	InFusion Textiles	4000- 110- 1200- 41111- 0000	4000- 110- 0000- 21020- 3100	1
IC Sales	InFusion Production	InFusion Farms	3200- 130- 1200- 41112- 0000	3200- 130- 0000- 21012- 5000	3

Intercompany Accounts Generated for Intercompany Credit Transactions

Transaction Type	Provider LE	Receiver LE	Provider Distribution	Provider AP Account Generated	Uses Rule No.
IC Sales	InFusion Farms	InFusion Textiles	3100- 100- 0000- 52330- 0000	3100- 100- 0000- 21011- 4000	2
IC Adjustments	InFusion Farms	InFusion Textiles	3100- 100- 1200- 52330- 0000	3100- 100- 0000- 21020- 4000	1
IC Sales	InFusion Production	InFusion Farms	5000- 120- 1300- 52345- 0000	5000- 120- 0000- 21020- 3200	1

This table displays the Receiver side of the transaction.

Transaction Type	Provider LE	Receiver LE	Receiver Distribution	Receiver AP Account Generated	Uses Rule No.
IC Sales	InFusion Farms	InFusion Textiles	4000- 100- 1200- 41111- 0000	4000- 100- 0000- 13013- 3100	4
IC Adjustments	InFusion Farms	InFusion Textiles	4000- 100- 1200- 41111- 0000	4000- 100- 0000- 13020- 3100	1
IC Sales	InFusion Production	InFusion Farms	3200- 130- 1200- 41112- 0000	3200- 130- 0000- 13012- 5000	3

## Reconcile Accounts

### Intercompany Reconciliation: Explained

Intercompany reconciliation provides you with reports to assist you with reconciling your intercompany receivables and intercompany payables accounts, and to identify any differences.

The main goal of the reports is to make it easy for you to identify either the receiver side or provider side of a transaction that has not been posted to the intercompany receivables or intercompany payables account.

The reports show the following intercompany lines:

- Intercompany receivables and intercompany payables lines generated by the intercompany balancing feature
- Intercompany receivables and intercompany payables lines generated for the provider and receiver of each intercompany transaction

The following are not included on the intercompany reconciliation reports:

- Ledger balancing lines generated when the primary balancing segment value is in balance but either the second balancing segment or the third balancing segment is out of balance
- Clearing company balancing lines

## Reconciliation Reports

The reconciliation reports show the Entered or Transaction amount of the accounting entries booked to the intercompany receivables and payables accounts for a pair of provider and receiver legal entities. Since the accounted amounts may be different if the conversion rates used for the intercompany receivables and intercompany payables are different, you can choose to run the reports using an additional currency and conversion rate that will convert all amounts into a common currency for comparison.

The intercompany reconciliation process starts with running the Extract Intercompany Reconciliation Data process. Choose from a variety of parameters to determine what data will appear on your reports. For example, choose the provider legal entity and receiver legal entity for which you want to run reconciliation.

Once the Extract Intercompany Reconciliation Data process has completed successfully, choose your request from the Oracle Business Intelligence Publisher (BI Publisher), BI Publishing Options list of values and view the Reconciliation Period Summary report. This report displays the intercompany receivables and intercompany payables balances in summary for a period, and any differences between them. Drill down on the hyperlinks to view the balances by source and then by journal lines. You have full drill down capabilities to the general ledger journal, subledger accounting entry and source receivables or payables transaction.

## Extract Intercompany Reconciliation Data

This process extracts data used to generate reports that can be viewed and utilized to assist with reconciliation.

You can run the report from the Intercompany Reconciliation task, and optionally schedule the report to run periodically.

### *Extract Intercompany Reconciliation Data Parameters*

#### **Ledger**

Ledger associated with the provider organization. Exclude secondary and reporting currency ledgers.

**Legal Entity**

Legal entity of the provider organization.

**Accounting Period**

Accounting period of the provider ledger.

**Ledger**

Ledger associated with the receiver organization.

**Legal Entity**

Legal entity of the receiver organization.

**Accounting Period**

Accounting period of the receiver ledger.

**Currency**

Currency for converting the accounted amount.

**Conversion Rate Type**

Conversion rate type for the additional currency.

**Conversion Rate Date**

Conversion rate date for the additional currency.



---



---

## Define Hyperion Financial Management Integration

### Oracle Hyperion Financial Management: Overview

Oracle Hyperion Financial Management, Fusion Edition is a comprehensive, web based application that delivers global collection, financial consolidation, reporting, and analysis in one highly scalable solution. Financial Management can be licensed and integrated with the Oracle Fusion Accounting Hub to provide expanded financial consolidation, reporting, and analysis functionality.

Financial Management supports these features:

- Comprehensive view of your enterprise financial information that provides key performance and operating metrics from global sources
- Close features that include intercompany reconciliations and a consistent set of data and business measures
- Powerful multidimensional analysis that identifies and reports new sources of profitability and cash flow at corporate, cost center, product, brand, customer, and channel levels
- Flexible what-if scenario management that dynamically consolidates and reports all financial budgets, forecasts and plans, producing new statements as assumptions and facts change
- High-volume, preformatted reports that deliver timely including financial information for internal management and external regulatory and government bodies

The following table provides installation and configuration activities and documentation references.

Activity	Documentation
Installation of Financial Management	Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide for Oracle Hyperion Enterprise Performance Management

Configuration of Financial Management including extended analytics	<ul style="list-style-type: none"> <li>• Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide for Oracle Hyperion Enterprise Performance Management</li> <li>• Oracle Hyperion Financial Management Administrator's Guide for Oracle Hyperion Financial Management</li> </ul>
Integration of Financial Management	Oracle Hyperion Financial Data Quality Management ERP Integration Adapter for Oracle Applications Administrator's Guide
License and integration with Oracle Hyperion Financial Close Management, which enables you to define, execute, and report on the interdependent activities of a financial close period. It provides centralized monitoring of all close process tasks and provides a visible, automated, repeatable system of record for running close processes.	Oracle Hyperion Financial Close Management Administrator's Guide for Oracle Hyperion Financial Close Management
License and integration with Oracle Hyperion Disclosure Management, which helps companies ensure the completeness and accuracy of external and statutory financial reports, and their transformation into specifically stylized statutory filings using eXtensible Business Reporting Language (XBRL).	Oracle Hyperion Disclosure Management User's Guide for Oracle Hyperion Disclosure Management

---

**Note**

There are now two integration points between Financial Management and the Oracle Fusion Governance, Risk, and Compliance solution set. Both integrations are delivered and deployed as Oracle Fusion Governance, Risk, and Compliance Blueprint solutions. See the Oracle Fusion Governance, Risk, and Compliance documentation.

---

## Integration with Hyperion Financial Management: Overview

For Oracle Cloud implementations, integrate with on-premise Oracle Hyperion Financial Management, Fusion Edition for advanced financial consolidations by manually exporting your balances from Oracle Fusion Accounting Hub and loading them to Oracle Hyperion Financial Management. Export your balances using either financial reports or Smart View.

For other implementations, Oracle Fusion Applications provides integration between Oracle Fusion Accounting Hub and Oracle Hyperion Financial Management through Oracle Financial Data Quality Management ERP Integrator adapter. To complete the post-installation setup for the ERP Integrator adapter, see Oracle Hyperion Financial Data Quality Management ERP Integrator Adapter for Oracle Applications Administrator's Guide.

---

---

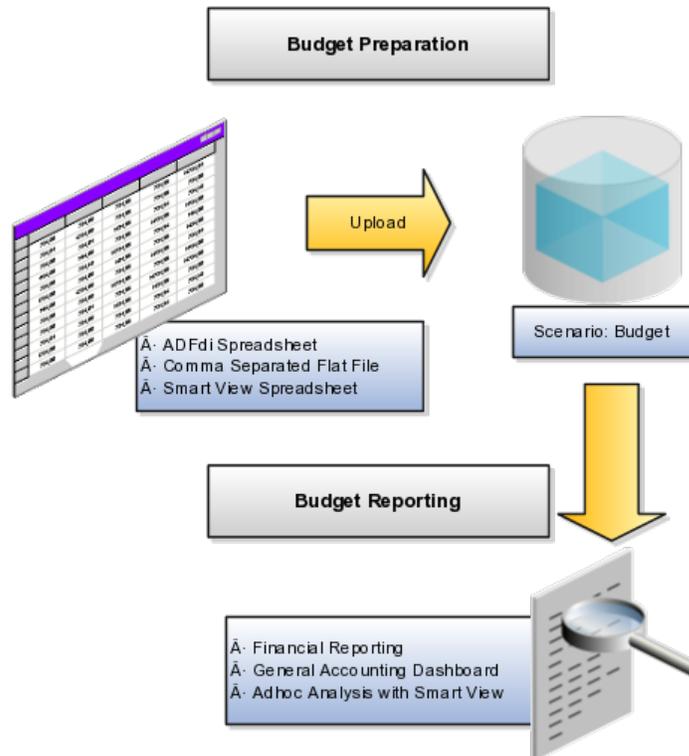
## Define Budget Configuration

### Budget Uploads: Overview

In Oracle Fusion General Ledger, you load budget data to perform variance reporting.

If you use a third party budgeting system or if you don't use a budgeting system, there are three ways to load budgets in the to the balances cube in the General Ledger.

- **Importing Budget Data from a Flat File:** Export budget data from your budgeting application to a comma separated values (csv) file. A sample xls template is provided in the Oracle Enterprise Repository (OER) for Oracle Fusion Applications. Use this template to prepare and generate flat files in a csv format. You can use Oracle Application Development Framework (ADF) Desktop Integrator correction worksheets to correct validation errors, delete rows with errors, and resubmit the corrected error rows.
- **Importing Budget Data from a Spreadsheet:** You can access the budget load spreadsheet from General Accounting Dashboard: Enter, load, and correct budget data in a ADF Desktop Integrator spreadsheet tool. Use this tool to prepare and load budget data for multiple ledgers and periods with a common chart of accounts instance. The list of values and the web picker help you pick valid values. This simplified data entry reduces errors and alerts you to errors as you enter the data in the spreadsheet. Error correction is done in the same spreadsheet.
- **Smart View:** Enter and load budget data in a Smart View spreadsheet. Use this tool to enter data for an account across multiple periods on a single row.




---

**Note**

Create reports in **Smart View** or **Financial Reporting** to verify that the budget data was loaded correctly.

---

## Importing Budget Data from a Flat File: Explained

Use the Upload Budgets processes to integrate budget information from other budgeting application such as Oracle Hyperion Planning, Fusion Edition. You can load your budget amounts to the General Ledger balances cube by populating the GL\_BUDGET\_INTERFACE table and running the Validate and Upload Budgets process. You can load budgets for multiple periods and for multiple ledgers with the same chart of accounts in a single load process. Note that the budget data is not loaded to the GL\_BALANCES table and only loaded to the balances cube for variance reporting purposes.

---

**Note**

You can load data to interface tables using predefined templates and the Load Interface File for Import scheduled process, which are both part of the External Data Integration Services for Oracle Cloud feature. For other implementations, optionally use this feature only if you have SFTP configured for it.

---

### Assigning Values for Columns in the GL\_BUDGET\_INTERFACE Table

You must enter values in all the columns of the interface table that require values, which includes all of the not null columns, in order for the budget import to be successful.

Enter values in the following required columns of the interface table:

Column Name	Values
RUN_NAME	Enter a name to identify the budget data set being imported.
STATUS	Enter the value NEW to indicate that you are bringing new budget data.
LEDGER_ID	Enter the appropriate ledger ID value for the budget amount. You can view the ledger ID for your ledgers in the Manage Primary Ledgers page. The ledger ID column is hidden by default, but you can display it from the View > Columns menu. If you enter multiple ledgers for the same run name, all of the ledgers need to share the same chart of accounts.
BUDGET_NAME	Enter the appropriate budget name value for the budget line. You define your budget names in the Accounting Scenario value set.
PERIOD_NAME	Enter the period name that you are loading the budget data for. Note that you can load budget data regardless of the period status.
CURRENCY_CODE	Enter the currency code for your budget.
SEGMENT1 to SEGMENT30	Enter valid enabled account value for each segment in your chart of accounts.
BUDGET_AMOUNT	Enter the amount in the ledger currency for account types, expense and assets.
OBJECT_VERSION_NUMBER	For Oracle Cloud implementations, leave this field blank as the application will automatically populate this when you load the data from the secure FTP server to the interface table. For other implementations, you can default the column with a value of 1.

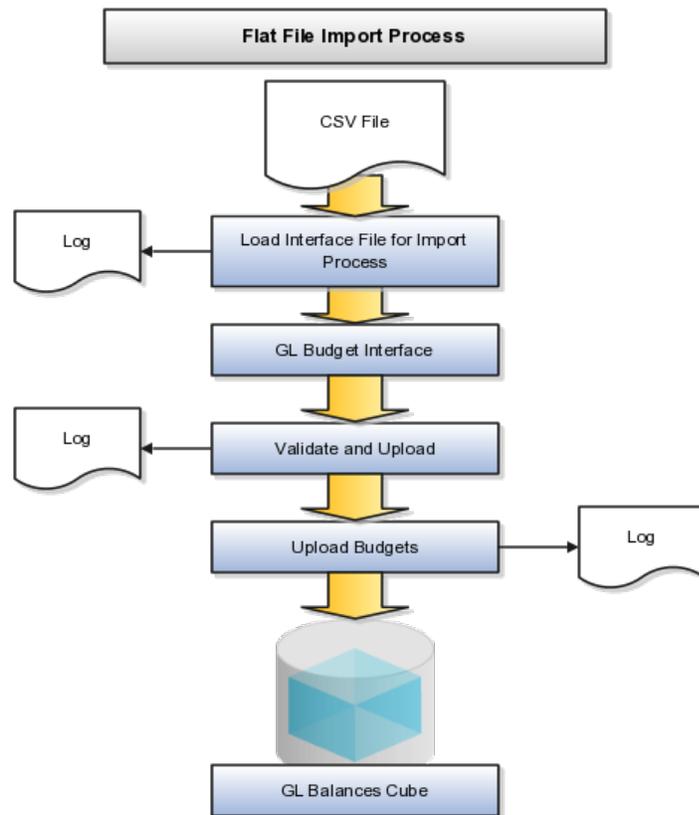
The following columns should be left as null as the budget import process uses these columns for internal processing or does not use them in the current release.

- CHART\_OF\_ACCOUNTS\_ID
- CODE\_COMBINATION\_ID
- ERROR\_MESSAGE
- CREATION\_DATE
- CREATED\_BY
- LAST\_UPDATE\_DATE
- LAST\_UPDATE\_LOGIN
- LAST\_UPDATED\_BY
- REQUEST\_ID
- LOAD\_REQUEST\_ID

## Loading Data to the Budget Interface Table: Explained

Load the budget amounts to the interface table by following the steps below.

1. Export budget data from your budgeting application to a comma separated values (csv) file. You can use the sample csv file or xls file that is provided in Oracle Enterprise Repository (OER) for Oracle Fusion Applications as a reference.
2. Upload the comma separated values (csv) file to the secure FTP server.
3. Select the parameters:
  - Select the process: General Ledger Validate and Load Budgets
  - Enter the name of the zipped CSV data file.
4. After the data is loaded to the interface table, you can run the Validate and Load Budgets process to load the budget amounts to the General Ledger balances cube.
5. Review the logs for validation errors. If there are validation errors, use an ADF Desktop Integrator (ADFdi) correction worksheet to download and correct the rows with errors. Then resubmit the data using ADFdi.



## Importing Budget Data from a Spreadsheet: Explained

Use Oracle Application Development Framework (ADF) Desktop Integrator to enter, load, and correct budget data. This functionality uses a new interface table

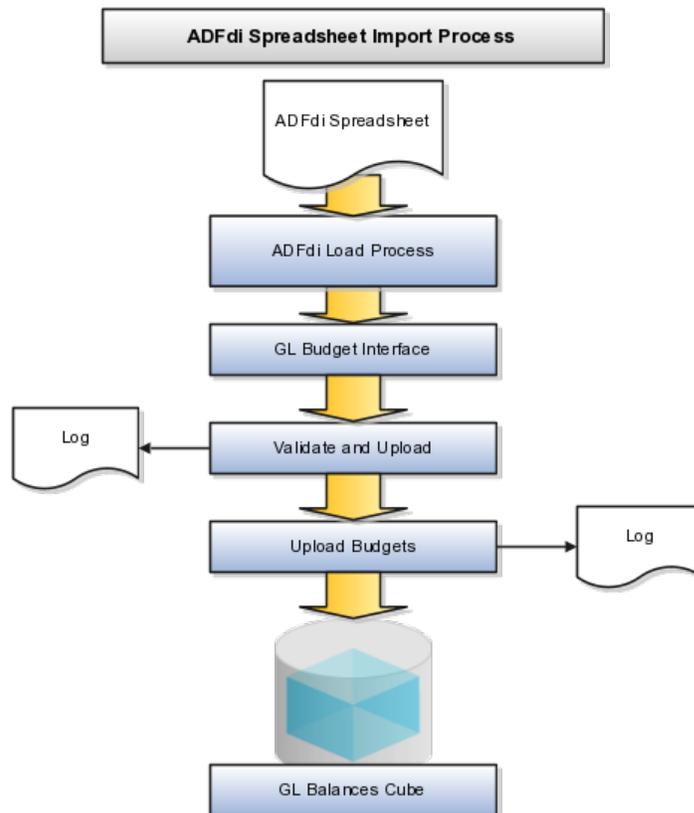
called the GL\_BUDGET\_INTERFACE and requires the duty role, Budget Entry Duty.

## Budget Import

Budget Import Oracle ADF Desktop Integrator import functionality is similar to the journal import sheet in Oracle Fusion General Ledger. You may use this tool to create and upload budget data. From the General Accounting Dashboard page, download the import worksheet.

The budget import uses the Accounting Scenario value set for the budget being loaded. The Run Name is used as an identifier for the imported data set. The Oracle ADF Desktop Integrator budget import functionality:

- Supports multiple ledgers but a single chart of accounts instance
- Allows multiple calendars and periods
- Supports entered currencies in addition to the ledger currency
- Contains user friendly lists of values
- Performs most validations on the worksheet
- Secures values by data access sets



---

### Note

The ADF Desktop Integrator spreadsheet contains a Record Status column that shows if the rows upload successfully or with errors. Corrections are done in the same spreadsheet as the entered data.

---

# Budget Correction with Oracle ADF Desktop Integrator: Explained

Oracle ADF Desktop Integrator correction functionality is similar to the journal correction sheet in Oracle Fusion General Ledger. You use this tool to correct the flat file import errors.

The correction spreadsheet functionality:

- Uses segment labels based on the data access set
- Contains user friendly lists of values
- Performs most validations on the worksheet
- Allows updating or marking the row for deletion.

## Correcting Data

To use the correction spreadsheet functionality perform the following steps:

1. From the **General Accounting Dashboard** page, you set the data access set and download the correction worksheet.
2. After the correction worksheet is downloaded, you query for the rows in error. Pick the run name for which there are validation errors and click on the **Search** button. This populates the budget rows in error.
3. Correct the rows in error or mark for deletion and submit the journal correction spreadsheet. Any errors will be reported on the worksheet.
4. If the row status indicates an error, double click on it to see the error details and take necessary action. You can use the list of values to quickly correct data that is in error.

## Creating Budget Data Security: Worked Example

You are in charge of your company's data security and have a need to secure budget data. You can secure budget data by using segment value security for the value set, Accounting Scenario. To meet this need for security, you create two data security polices.

The following budget version values from the accounting scenario value set need to be secure:

- Original Budget
- Revised Budget
- Forecast Q1
- Forecast Q2
- Forecast Q3

- Forecast Q4

The following data security needs to be configured:

- Policy 1: The financial controller for Vision Foods USA data access set needs access to all budget versions.
- Policy 2: The general accounting manager for Vision Foods USA data access set needs access to the budget versions listed below:
  - Original Budget
  - Revised Budget

In this example, launch the task Define Budget Scenarios to open the value set page from the Setup and Maintenance work area.

1. Click the Edit icon.
2. Check Security Enabled.
3. Enter the Data Security Resource name (if not populated): ACCOUNTING\_SCENARIO.
4. Click on Edit Data Security to set up the data policies.

## Setting up Policy 1: Full Access to All Budget Versions

1. Click on the Policy tab.
2. Click on Create.
3. On the General Information tab, enter Name, Start and End Dates, and Module equal to General Ledger.
4. On the Role tab, add one or more roles. In this case add the Chief Finance Controller for Vision Foods USA.
5. On the Rule tab, select Row Set equal to All Values.
6. Save and Submit the policy.

## Setting up Policy 2: Access to One or More Budget Versions

1. Click on the Condition tab to create a filter for the multiple values.
2. Set Match to Any to invoke OR Boolean operator. If you set Match to All, AND Boolean operator is used.
3. Enter two rows with Column Name: Values. Operator: Equal to, and Value: Original Budget on first line and Revised Budget on the second line. Note that you can also use the tree operators.
4. Click on the Policy tab.
5. Click on Create.
6. On the General Information tab, enter Name, Start and End Dates, and Module equal to General Ledger.
7. On the Role tab, add one or more roles. In this case add the General Accounting Manager for Vision Foods USA.

8. On the Rule tab, select Row Set equal to Multiple Values.
9. Select the Condition Filter
10. Save and Submit the policy.

---

**Note**

You can also set up single access by using a policy with Row Set equal to Single value. You need to use SQL to get the value ID and enter the value ID.

---

## Oracle Hyperion Planning: Overview

Oracle Hyperion Planning, Fusion Edition is a comprehensive, web based, budgeting and planning application. Planning can be licensed and integrated with the Oracle Fusion Accounting Hub to provide expanded budgeting and planning functionality.

Oracle Hyperion Planning, Fusion Edition:

- Delivers global collection of data and financial consolidation
- Enables entry, analysis, and reporting on data, including personalizing data entry forms
- Facilitates budgeting collaboration, communication, and control across multidivisional global enterprises
- Drives the planning process
- Provides a framework for perpetual planning, with attention to managing volatility and frequent planning cycles
- Promotes modeling using complex business rules and allocations
- Integrates with Smart View so you can design worksheets in Microsoft Excel to enter, format, analyze, and report on data

The following table provides installation and configuration activities and documentation references.

Activities	Documentation
Installation of Planning	Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide for Oracle Hyperion Enterprise Performance Management
Configuration of Planning	Oracle Hyperion Planning, Fusion Edition Administrator's Guide for Oracle Hyperion Planning
Integration of Planning	Oracle Hyperion Financial Data Quality Management ERP Integration Adapter for Oracle Applications Administrator's Guide

## Integration with Hyperion Planning: Overview

For Oracle Cloud implementations, integrate with on-premise Oracle Hyperion Planning for advanced budgeting by loading actual balances from Oracle Fusion Accounting Hub to Oracle Hyperion Planning so you can use the actual data in the budgeting process. You can also load budget data from Oracle Hyperion Planning to Oracle Fusion Accounting Hub through the Budget Interface to perform budget variance reporting within Oracle Fusion Accounting Hub.

For other implementations, Oracle Fusion Applications provides integration between Oracle Fusion Accounting Hub and Oracle Hyperion Planning through Oracle Financial Data Quality Management ERP Integrator adapter. To complete the post-installation setup for the ERP Integrator adapter, see Oracle Hyperion Financial Data Quality Management ERP Integrator Adapter for Oracle Applications Administrator's Guide.



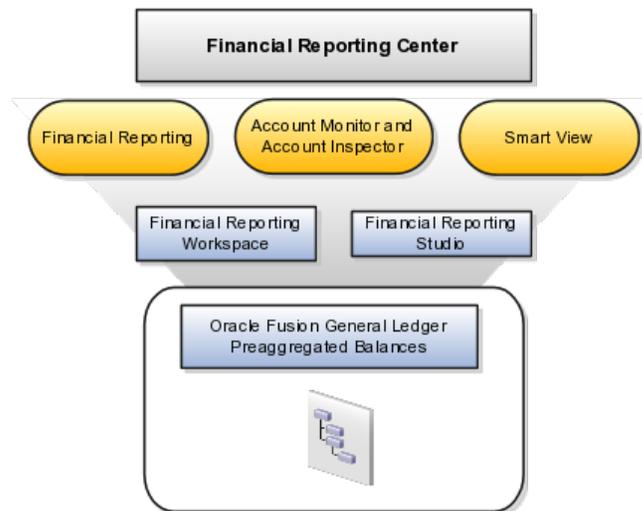
---

## Define Financial Reporting

### Financial Reporting Center: How It Works

The Oracle Fusion Financial Reporting Center provides functionality for reporting on Oracle Fusion General Ledger balances. It provides secure, self-service access to reports that use real time account information.

You can design traditional financial report formats such as balance sheets, profit and loss statements, and cash flow reports. You can also design nontraditional formats for financial or analytic data that include text and graphics.



### Components

Financial Reporting Center is comprised of numerous components:

- Financial Reporting: Financial users and analysts access live reports and books or published snapshot reports and books from previously scheduled batches in a variety of formats. Other functionality includes:
  - Refreshing report data using runtime points of view or parameters
  - Drill through capability from parents to other parents
  - Drill down to detail balances, journal lines, and subledger transactions.

- Oracle Hyperion Smart View, Fusion Edition: Financial analysts view, import, manipulate, distribute, and share data from your Oracle Fusion General Ledger balances in Microsoft Excel.
- Account Monitor and Account Inspector: Financial analysts monitor and track key account balances in real time at every level of your dimensions and hierarchies. These tools provide multidimensional account analysis and drill down capability.
- Workspace: Reporting administrators create, open, save, and delete folders and store report objects, reports, and snapshot reports.
- Oracle Hyperion Financial Reporting Studio, Fusion Edition: Report authors use an object-oriented graphical report layout with report objects, such as text boxes, grids, images, and charts, to design reports.

## Setting Up Your Financial Reporting Center: Critical Choices

Oracle Fusion Financial Reporting Center is a powerful tool for accessing, designing, and presenting financial reports and analytic data. The critical choices needed to configure and install the components in Financial Reporting Center consist of:

- Configuring Financial Reporting Center
- Installing and configuring Financial Reporting Studio (Performed by the end user.)
- Installing Smart View (Performed by the end user.)
- Configuring Workspace Database Connection (Performed by the administrator.)
- Installing and Configuring Financial Reporting Print Server (Performed by the administrator.)

### Configuring Financial Reporting Center

You have access to the reports through the folder structure in the Financial Reporting Center and Workspace. Your administrators define the folder structure in Workspace taking into consideration your company's security requirements for folders and reports, as well as report distribution requirements for financial reporting batches. Security can be set on folders and reports from Workspace. You are granted access to the folders and reports you need to view.

### Installing and Configuring Financial Reporting Studio

Oracle Hyperion Financial Reporting Studio is client-based software. If you access Oracle Fusion Applications from Oracle Cloud, you connect to the Financial Reporting Studio through a Windows Remote Desktop Connection. Otherwise, report authors download the installation files for Financial Reporting Studio from **Workspace** by clicking on **Navigator > Tools > Download Desktop Integrator Installer**. After completing the installation, launch the Financial Reporting Studio. You will need to provide your user ID, password, and the Server URL. Derive the Server URL information by following the steps below:

1. Open **Navigator > Financial Reporting Center > Workspace** .

2. Edit the **Workspace URL** and remove workspace/index.jsp.
3. Below are two examples of Server URLs:
  - If the Workspace URL is https://fusionsystemtest-p-external-bi.us.oracle.com/workspace/index.jsp, the Server URL is https://fusionsystemtest-p-external-bi.us.oracle.com.
  - If the if the Workspace URL is https://fusionsystemtest-p-external-bi.us.oracle.com:10622/workspace/index.jsp, the Server URL is https://fusionsystemtest-p-external-bi.us.oracle.com:10622.

---

**Note**

For Oracle Cloud based environments, the Workspace URL includes the port number.

- 
4. Copy the modified URL to the Server URL field.

**Installing Smart View**

Smart View is an Excel add-in that must be loaded to each client. To download the installation files from **Workspace** click the **Navigator > Tools > Download Desktop Integrator Installer**.

---

**Note**

Since Smart View is an add-in to Microsoft Office products, you can install Smart View only on a Windows operating system.

---

Once Smart View is installed, it must be configured to connect to Oracle Fusion Applications. This is done using the Smart View Shared Connections URL. You can derive the **Shared Connections URL** by following the steps below:

1. **Launch Workspace** from the **Financial Reporting Center**.
2. Edit the **Workspace URL**, for example, if the **Workspace URL** is https://fusionsystemtest-p-external-bi.us.oracle.com/workspace/index.jsp. Remove index.jsp and add SmartViewProviders at the end of the URL.

---

**Note**

This is another example for a Cloud based environment: If the Workspace URL is https://efops-rel5st4-cdrm-external-bi.us.oracle.com:10622/workspace/index.jsp, the Shared Connections URL is https://efops-rel5st4-cdrm-external-bi.us.oracle.com:10622/workspace/SmartViewProviders.

- 
3. Copy the URL.
  4. Launch Excel.
  5. Navigate to the **Smart View** menu > **Options > Advanced**.
  6. Paste the URL in the **Shared Connections URL** field.
  7. Click on the **OK** button.

For more information on configuring Smart View client for users, see Oracle Hyperion Smart View for Office User's Guide for Oracle Hyperion Smart View.

To connect Oracle Fusion General Ledger Balances cubes in Smart View:

1. Open Smart View from your **Start menu > Programs > Microsoft Office > Microsoft Excel 2007**.
2. Go to the **Smart View** menu > **Open**, in the **Start** on the ribbon > click on **Smart View Panel** that appears in the drop down box under the ribbon. This launches a task pane.
3. Click on the **Shared Connections** button on the task pane.
4. Sign in with your user name and password.
5. Click on the **Select Server to proceed** drop down box.
6. Select **Oracle Essbase** from the list of shared connections.
7. Click the **Expand** to expand the list of cubes.
8. Expand your cube (name of your chart of accounts).
9. Click on **db**. A list of functions appears on the bottom of the panel.
10. Click the **Ad hoc analysis**.

---

#### Note

You need to perform these steps only once for a new server and database.

---

To set how the name and alias of the Essbase database appears:

1. Click on the **Options** on the ribbon > select the **Member Options** > select **Member Name Display**.
2. Set one of these three options:
  - **Distinct Member Name:** Only shows the full Essbase distinct path.
  - **Member Name and Alias:** Shows both the member name and the alias.
  - **Member Name Only:** Shows only the member name.

---

#### Note

The following Smart View features are not supported in Oracle Fusion General Ledger. For all other documentation, refer to the **Oracle Hyperion Smart View for Office User's Guide** for Oracle Hyperion Smart View.

- **Smart Slice:** Not supported.
  - **Query Designer:** Currently has a bug.
- 

### Configuring Workspace Database Connections

Administrators need to create database connections from Workspace so users can access the cubes from Workspace and Financial Reporting Studio.

---

#### Note

Ledger setup has to be completed before the database connection can be created. Oracle Fusion General Ledger balances cubes are created as part of ledger setup. There is a separate cube for each combination of chart of accounts and accounting calendar. A database connection is needed for each cube.

---

Steps to define a database connection are:

1. Start at the **Navigator** by selecting **Financial Reporting Center**.
2. From the **Financial Reporting Center** task panel select **Launch Workspace**.
3. From within **Workspace** select the **Navigator** menu > **BI Catalog**.
4. Select **Tools** menu > **Database Connection Manager**.
5. Select **New** button.
6. Enter a user friendly name for the **Database Connection Name**.
7. Enter Essbase as the **Type**, your server, user name, and password.
8. Select **Application** (cube) and **Database** from the list of values. Expand the **Application** name to see the related **Database**, for example, db.
9. Click the **OK** button twice to save your selections.
10. Click **Close** button in the **Database Connection Manager** window to save your connection.

For more information on configuring Essbase database connections in Workspace see: **Oracle Essbase Database Administrator's Guide for Oracle Essbase**.

---

#### Note

The database connection is available in both Workspace and Financial Reporting Studio. Optionally, it can be setup in Financial Reporting Studio when putting grids on a report. This should only be done by an administrator.

---

## Installing and Configuring Financial Reporting Print Server

To enable users to view and print their reports, administrators need to install and configure the Financial Reporting Print Server available with Oracle Business Intelligence. However, check for the following prerequisites before the installation:

- Install the Financial Reporting Studio.
- If you have an earlier release of the Financial Reporting Print Server, stop and disable the Financial Reporting Print Server Windows service, and delete HRPrinter1-5 from Windows printers.
- Ensure that a supported 32-bit version of Ghostscript is installed.

Perform the following steps to complete the installation.

1. Obtain a Windows machine (either 32-bit or 64-bit) to install and configure the Financial Reporting Print Server.
2. Sign in to Oracle Business Intelligence Enterprise Edition.

3. Open **Navigator > Financial Reporting Center > Workspace** .
4. If you have not already installed Financial Reporting Studio, you can download and install the Financial Reporting Studio from **Navigator > Tools > Download Desktop Integrator Installer**.
5. Follow the instructions to complete the installation. By default, the Financial Reporting Studio is installed in the Program Files/Oracle/FinancialReportingStudio directory.
6. In the Program Files\Oracle\FinancialReportingStudio\products\financialreporting\install\bin\ directory, search for the FRSetupPrintServer.properties file and open it with a text editor.
7. For the following attributes, specify the Financial Reporting Server URL and administrator credentials used to register the Financial Reporting Print Server:

```
FRWebServer=http://<name of the server>:<port number>
AdminUser= <Administrator user name>
AdminPassword=<password>
```

---

### Note

The server URL specified here is the same one used when connecting from the Financial Reporting Studio.

---

8. To configure and register the Financial Reporting Print Server, in the Windows command prompt, navigate to FinancialReportingStudioInstallationLocation/products/financialreporting/install/bin/ and run the following command:

```
FRSetupPrintServer.cmd
```

The Financial Reporting Print Server is now installed and configured for use. You should be able to view the reports in PDF. However, to ensure that everything is running as required, verify the following:

- Financial Reporting Print Server service has been created and has started automatically.
- The FRPrintLogging.log file is created in the directory Program Files\Oracle\FinancialReportingStudio\diagnostics\logs\FinancialReporting.
- The `PrintServers` property in the Financial Reporting MBeans in JConsole displays the Financial Reporting Print Server machine and port (the default port is 8297). This property may show multiple Financial Reporting Print Servers if more than one have been configured.
- The server name and port of the Financial Reporting Print Server can be reached from the Financial Reporting Web Application server machine(s).

## Oracle Fusion General Ledger Predefined Reports

Oracle Fusion General Ledger provides predefined reports that are used in the close process and to verify setup of the accounting configuration.

### Oracle Business Intelligence Publisher

The reports classified as Oracle Business Intelligence Publisher (BI Publisher) are scheduled and run from the **Scheduled Processes** work area on the Navigator menu.

**Oracle Fusion Translational Business Intelligence**

All the reports including those classified as Oracle Fusion Transactional Business Intelligence (BI) are accessed from the **Reports and Analytics** pane on the **Navigator** menu or from dashboards, where the reports are saved. The report links in the **Reports and Analytics** pane open in the **Oracle Business Intelligence Catalog** where reports can be edited, printed, and reviewed.

The following tables are the lists of predefined reports by type.

**Account Analysis Reports**

Process Name	Display Name	Description	Classification
GLACTANL2	Account Analysis	Prints balances by account segment and a secondary segment for each journal entry, and lists the subledger document number for transactions imported from subledgers.	Oracle Fusion Transactional Business Intelligence (BI)
GLRFCLD	General Ledger Report	Provides journal information to trace each transaction back to its original source. Prints a separate page for each balancing segment value. For each journal line, prints the account affected, the concatenated description, the journal line amount, and the beginning and ending account balance. Additionally, for each journal line, prints journal details including source, category, journal name, and effective date. Lists accounts in ascending order by account segment value, and prints a CR next to credit amounts.	Oracle Business Intelligence Publisher (BI Publisher)
GLWACCTR	Account Analysis for Contra Account Report	Prints balances by account segment and a secondary segment, lists the contra account for each journal entry, and lists the subledger document number for transactions imported from subledgers. Print this report by date range, accounting flexfield range, contra account, and amount range.	BI Publisher

GLXAVADT	Average Balance Audit Account Analysis Report	Displays the detail account activity which created the aggregate balances and related average balances. Used to research how average balances are calculated for an account. Displays daily average balance information for the selected accounts for the specified range of dates. Contains parameters such as the as-of reporting date, average balance type (period, quarter, or year average-to-date), and account ranges.	BI Publisher
----------	---	--	--------------

### Journal Reports

Process Name	Display Name	Description	Classification
GLJRNL	Journals Report	Provides journal activity for a given period or range of periods, balancing segment value, currency, and range of account segment values. Prints the accounting date, category, journal name, reference, journal batch name, entered debit or credit amounts, net balance, and account total for each journal. In addition, a total is provided for each balancing segment and a grand total is provided for all the activity included.	BI Publisher
GLRXVCJ	General Journals Report	Provides journal activity for a given period or range of periods, balancing segment value, currency, and range of account segment values.	Oracle Fusion Transactional (BI)

GLYRLJRE	Journals Batch Summary Report	Lists posted journal batches for a particular ledger, balancing segment value, currency, and date range. Provides information on actual balances for your journal batches, source, batch, and posting dates, total entered debits and credits, and sorts the information by journal batch within each journal entry category. In addition, totals are provided for each journal category and a grand total for each ledger and balancing segment value combination included in your report. Does not report on budget or encumbrance balances.	BI Publisher
GLRXCKJ	Journals Check Report	Provides information on manually entered journals prior to posting, including field by field, all data entered into the system or data imported from external sources.	BI Publisher
GLRXDBJ	Journals Day Book Report	Provides posted journal entries and journal details chronologically by accounting date for a specified range of dates, journal source, and journal category. For each accounting date, journal entries are sorted by document number. Prints the accounting date, document number, journal entry name, journal source and category, subledger document name and number, currency, and conversion rate. For each journal line, it also prints the line number, account segment value and description, functional debit and credit amounts, description, and cost center segment value.	BI Publisher

**Trial Balance Reports**

Process Name	Display Name	Description	Classification
GLTRBAL	Trial Balance Report	Provides summarized actual account balances and activity by ledger, balancing segment, and account segment value.	BI Publisher and Oracle Fusion Transactional (BI)
GLXAVTRB	Trial Balance - Average Balances	Provides a listing of ending balances and average balances for selected accounts based on an as of date specified. Print the ledger currency or foreign-entered balances. In addition, displays period, quarter, and year average-to-date balances. Request additional information by specifying balancing segments and account ranges.	BI Publisher

### Reconciliation Reports

Process Name	Display Name	Description	Classification
GLReconciliationReport	Cash to General Ledger Reconciliation Report	Extracts cash management and general ledger accounting and transactional data for reconciling cash management to the general ledger.	BI Publisher
Payables to Ledger Reconciliation Report	Payables to Ledger Reconciliation Report	Provides both summarized and detailed reconciling data for review. Shows payables and accounting beginning and ending balances, as well as summarized activity for the period and how this activity was accounted.	Oracle Fusion Transactional (BI)
Receivables to Ledger Reconciliation Report	Receivables to Ledger Reconciliation Report	Provides reconciliation of receivables data to the general ledger. Shows receivables and accounting beginning and ending balances, as well as summarized activity for the period and how the activity was accounted.	Oracle Fusion Transactional (BI)

### Chart of Accounts Reports

Process Name	Display Name	Description	Classification
GLXBSVA	Balancing Segment Value Assignments Report	Reports on the assignment of primary balancing segment values to legal entities and ledgers across accounting setups. Allows quick identification of overlapping balancing segment value errors and reviews of any unassigned values. The application does not check for overlapping balancing segment values online.	BI Publisher
GLXCOCRR	Chart of Accounts Mapping Rules Report	Provides both the segment and account rules defined for a specific chart of accounts mapping. If the mapping has account rules, prints each subsidiary account range and the parent account into which it maps. If the mapping has segment rules, prints the rule name and the parent and subsidiary segments. If the mapping has a rollup range rule, prints each subsidiary segment value range and its corresponding parent segment value.	BI Publisher

To run BI Publisher reports, use the following steps the Scheduled Processes work area in the Navigator menu.

1. Click the **Schedule New Process** button
2. Search on the Process Name.
3. Enter the desired parameters.
4. Enter the desired process options and schedule.
5. Click Submit.



---

---

# Define Extensions for Fusion Accounting Hub

## Define Custom Enterprise Scheduler Jobs for Fusion Accounting Hub

### Managing Job Definitions: Highlights

Oracle Enterprise Scheduler jobs are run in Oracle Fusion Applications to process data and, in some cases, to provide report output. A job definition contains the metadata that determines what the job does and what options are available to users who run the job. You can create and maintain job definitions for use in Oracle Fusion Applications.

Managing job definitions is fully described in the Oracle Fusion Applications Administrator's Guide. As you read content from that guide, note that the guide describes managing Oracle Enterprise Scheduler, including job definitions, from Oracle Enterprise Manager Fusion Applications Control. You can also access job definitions by starting in the Setup and Maintenance Overview page and searching for the Enterprise Scheduler job tasks for your applications.

#### Selecting the Appropriate Implementation Task

Each Enterprise Scheduler job definition task uses one Java EE application, which is referenced in the task name. You must use the right task because, to access the product job definition that you want to view or work on, the view objects must be contained in the application. If you do not select the right task, then the job definition will not be displayed properly or function correctly. The application name is usually the same as the product that the job definition belongs to, but not always.

- For example, the Oracle Fusion Payables Java EE application contains the Oracle Fusion Expenses product view objects. To create or maintain a job definition for use in Expenses, you select the Manage Custom Enterprise Scheduler Jobs for Payables and Related Applications task.
- In another example, the Oracle Fusion Payments product view objects are contained in both Oracle Fusion Payables and Oracle Fusion Receivables

Java EE applications. You need to select the task appropriate to the job definition for Payments. Use the Manage Custom Enterprise Scheduler Jobs for Receivables and Related Applications task if the job is for receivables functionality, or the Manage Custom Enterprise Scheduler Jobs for Payables and Related Applications task if the job is for payables functionality.

- Use the task description to see the products that correspond to the Java EE application specified in the task name. For example, the description for the Payables task lists Oracle Fusion Payables, Assets, Expenses, and Payments.
  - You can view task descriptions in the help window for the task, if any, or in the generated setup task lists and tasks report from the Getting Started page.
  - If you have access to the Manage Task Lists and Tasks page, you can also open the details for specific tasks to see the description.
  - For general information about product and Java EE application relationships, use Oracle Enterprise Manager Fusion Applications Control (Fusion Applications Control).

See: Topology Section

### **Viewing and Creating Job Definitions**

- You can access predefined and custom job definitions.

See: Viewing Job Definitions

- You can create jobs based on Java, PL/SQL, or any other supported technology.

See: Creating a Job Definition

- If you are using the Setup and Maintenance work area, then the Enable submission from Enterprise Manager check box is available for the job definition.
  - If you do not select this check box, then the job cannot be run from Enterprise Manager.
  - If you select this check box, then you can define parameters for this job definition only in Enterprise Manager. Save the rest of your work on the job definition, and then go to Enterprise Manager if you need to define parameters.

## **Managing List of Values Sources: Highlights**

A list of values source for Oracle Enterprise Scheduler job definitions determines where a list of values comes from and what the specific values are. These lists of values are used in parameters and user properties of job definitions. For example, you can use a source of country values for a Country job parameter.

Managing list of values sources is fully described in the Oracle Fusion Applications Administrator's Guide. As you read content from that guide, note that the guide describes managing Oracle Enterprise Scheduler, including

list of values sources, from Oracle Enterprise Manager Fusion Applications Control. You can also access list of values sources by starting in the Setup and Maintenance Overview page and searching for Enterprise Scheduler job tasks.

### **Registering and Searching for List of Values Sources**

- Create list of values sources to register them for use in job definitions.

See: Registering Sources for Lists of Values

- Search for list of values sources to edit or delete, or to make sure a particular source does not already exist before you create it.

See: Searching for List of Value Sources



---

# Glossary

**account rule**

The rule that builds the account on a subledger journal entry. It can be used to derive complete accounts or a segment value. Conditions can be defined within a rule so that a different account is used based on particular attributes of a transaction.

**accounting attribute**

Predefined fields that map to components of subledger journal entries. Sources are assigned to accounting attributes.

**accounting calendar**

Period of time, extending from one date to another, defined to quantify the fiscal position of the business. It can be divided into accounting periods.

**accounting class usage**

Grouping of accounting classes to be used in reporting or processing based on subledger journal entries.

**accounting event class**

Categories that classify transaction types and group event types for accounting rules.

**accounting event type**

Represents a business operation that may have an accounting impact.

**accounting method**

A set of journal entry rules which determine how a subledger journal entry is to be created for each event class or event type.

**accounting period**

The fiscal period used to report financial results, such as a calendar month or fiscal period.

**API**

Abbreviation for application programming interface.

**application feature**

A standardized functionality that is available to implemented.

**application identity**

Predefined application level user with elevated privileges. An application identity authorizes jobs and transactions for which other users are not

authorized, such as a payroll run authorized to access a taxpayer ID while the user who initiated the job is not authorized to access such personally identifiable information.

**AutoPost criteria sets**

A grouping of options and submission frequencies used to select journal entries for automatic posting.

**balancing segment**

A chart of accounts segment used to automatically balance all journal entries for each value of this segment.

**business object**

A resource in an enterprise database, such as an invoice or purchase order.

**business unit**

A unit of an enterprise that performs one or many business functions that can be rolled up in a management hierarchy.

**chart of accounts**

The account structure your organization uses to record transactions and maintain account balances.

**clearing company**

The intercompany clearing entity used to balance the journal.

**constant**

Holds the numeric value used to evaluate numeric conditions in Contract Expert rules. A constant permits you to reset the conditions of many rules with just one edit.

**context**

A grouping of flexfield segments to store related information.

**context segment**

The flexfield segment used to store the context value. Each context value can have a different set of context-sensitive segments.

**context-sensitive segment**

A flexfield segment that may or may not appear depending upon a context such as other information that has been captured. Context-sensitive segments are custom attributes that apply to certain entity rows based on the value of the context segment.

**conversion rate**

Ratio at which the principal unit of one currency can be converted into another currency.

**corporate rate type**

Rate you define to standardize rates used in conversion of one currency to another over a period of time. This rate is generally a standard market rate determined by senior financial management for use throughout the organization.

**cost center**

A unit of activity or group of employees used to assign costs for accounting purposes.

**cost organization**

A grouping of inventory organizations that indicates legal and financial ownership of inventory, and which establishes common costing and accounting policies.

**country holding company**

A legal entity that acts on behalf of several divisions within an enterprise, and is the legal employer in a country.

**cube**

A block of data that contains three or more dimensions. An Essbase database is a cube.

**department**

A division of a business enterprise dealing with a particular area of activity.

**description rule**

The rule that defines the description content that appears on the subledger journal header and line.

**descriptive flexfield**

Customizable expansion space, such as fields used to capture additional descriptive information or attributes about an entity, such as customer cases. Information collection and storage may be configured to vary based on conditions or context.

**division**

A business-oriented subdivision within an enterprise. Each division is organized to deliver products and services or address different markets.

**document sequence**

A unique number that is automatically or manually assigned to a created and saved document.

**enterprise**

An organization with one or more legal entities under common control.

**ESS**

Acronym for Enterprise Storage Server. An application that optimizes data storage.

**financial reporting book**

Comprised of reports and other documents such as text, PDF, PowerPoint, Excel and Word files. When run, the report data is dynamically retrieved from the database; the snapshot data remains static.

**fixed rate type**

Rate you set between two currencies that remains constant. For example, a rate set between the euro currency and each Economic and Monetary Union (EMU) currency during the conversion to the euro currency.

**flexfield**

Grouping of extensible data fields called segments, where each segment is an attribute added to an entity for capturing additional information.

**flexfield segment**

An extensible data field that represents an attribute on an entity and captures a single atomic value corresponding to a predefined, single extension column in the Oracle Fusion Applications database. A segment appears globally or based on a context of other captured information.

**interface table**

A database table used for transferring data between applications or from an external application or data file.

**item master**

A collection of data that describes items and their attributes recorded in a database file.

**job**

A generic role that is independent of any single department or location. For example, the jobs Manager and Consultant can occur in many departments.

**journal**

An element of a journal entry consisting of the name, accounting date, category, ledger, and currency for single currency journal entries. Used to group journal lines.

**journal batch**

An element of a journal entry consisting of the name, source, and accounting period. Used to group journals for processing and easier querying.

**journal category**

A name used to group journal entries with similar characteristics, such as adjustments, accruals, or reclassifications.

**journal entry**

Point of entry of business transactions into the accounting system. Chronological record, with an explanation of each transaction, the accounts affected, and the amounts to increase or decrease each account.

**journal line**

An element of journal entries consisting of account combinations and credit or debit amounts. Optionally, contains statistical quantities, currency information for multicurrency journals, and additional information.

**journal line rule**

A rule that includes options to convert transactional data into a subledger journal line. A condition can be defined within a rule so that the rule is only used based on particular attributes of a transaction.

**journal source**

A name that indicates the origin of journal entries, such as payables, receivables, or manual. Used as an attribute in automatic posting and journal import processes.

**key flexfield**

Configurable key consisting of multiple parts or segments, each of which may be meaningful individually or in combination with the others. Key flexfields are commonly implemented to represent part numbers and account numbers.

**legal authority**

A government or legal body that is charged with powers such as make laws, levy and collect fees and taxes, and remit financial appropriations for a given jurisdiction.

**legal employer**

A legal entity that employs people.

**legal entity**

An entity is identified and given rights and responsibilities under commercial law, through the registration with the country's appropriate authority.

**legal jurisdiction**

A physical territory, such as a group of countries, single country, state, county, parish, or city, which comes under the purview of a legal authority.

**legal reporting unit**

The lowest level component of a legal structure that requires registrations. Used to group workers for the purpose of tax and social insurance reporting or represent a part of your enterprise with a specific statutory or tax reporting obligation.

**legislative data group**

A means of partitioning payroll and related data. At least one legislative data group is required for each country where the enterprise operates. Each legislative data group is associated with one or more payroll statutory units.

**line of business**

Set of one or more highly related products which service a particular customer transaction or business need. Refers to an internal corporate business unit.

**mainline**

A branch of data that serves as a single source of truth.

**mapping set**

Maps a combination of input source values to specific output values. A mapping set can have a segment, account, or value set as output. The output value of a mapping set is used to derive accounts or segments in account rules.

**natural account**

Categorizes account segment values by account type, asset, liability, expense, revenue, or equity, and sets posting, budgeting, and other options.

**natural account segment**

A chart of accounts segment used to categorize your accounting transactions by account type: asset, liability, owner's equity, revenue, or expense.

**offering**

A comprehensive grouping of business functions, such as Sales or Product Management, that is delivered as a unit to support one or more business processes.

**payroll statutory unit**

A legal entity registered to report payroll tax and social insurance. A legal employer can also be a payroll statutory unit, but a payroll statutory unit can represent multiple legal employers.

**period type**

Is the shortest period range that must be available for incentive compensation processing, for example monthly. Associate the period type with a calendar and define incentive compensation periods (for example, Jan-10, Feb-10) for a calendar based on it.

**PL/SQL**

Abbreviation for procedural structured queried language.

**point of view**

User selected dimensions that are not included in the grids at the row, column or page levels for a particular report. Only these dimensions can be overridden at run time, unless user also specifically defined Prompt for the dimensions on the grid.

**position**

A specific occurrence of one job, fixed within one department, also often one location. For example, the position Finance Manager is an instance of the job Manager in the Finance Department.

**post**

Update account balances by the recorded debit or credit amount in each journal entry to the related accounts in the general ledger.

**primary balancing segment value**

A segment value used to represent a legal entity in the chart of accounts and automatically balance all intercompany and intracompany transactions and journal entries.

**primary ledger**

Main record-keeping ledger.

**process category**

Group of one or more logically related event classes. Can be used to restrict which events are processed by the Create Accounting process.

**profile option**

User preferences and system configuration options consisting of a name and a value, that can be set at hierarchical levels of an enterprise. Also called a profile or user option.

**reference data**

Data in application tables that is not transactional and not high-volume such as sales methods, transaction types, or payment terms, and can be shared and used across organizational boundaries.

**reference data set**

Contains reference data that can be shared across a number of business units or other determinant types. A set supports common administration of that reference data.

**reference object**

Standardized data model containing reference information owned by other subledger applications and used by the Create Accounting process to create subledger journal entries from accounting events.

**registration**

The record of a party's identity related details with the appropriate government or legal authorities for the purpose of claiming and ensuring legal and or commercial rights and responsibilities.

**revaluation**

The process of adjusting asset, liability, or income statement accounts that may be materially understated or overstated due to a significant fluctuation in the conversion rate between transaction and realization dates.

**sandbox**

A runtime session that commits changes out of reach of mainline users.

**secondary ledger**

An optional, additional ledger that is associated with the primary ledger for an accounting setup. Secondary ledgers can represent the primary ledger's data in another accounting representation that differs in chart of accounts, accounting calendar, currency, subledger accounting method and ledger processing options.

**set**

Reference data that is organized into groups appropriate to organizational entities, to enable reference data sharing.

**source**

Contextual and reference information from subledger applications. This information is used in conjunction with accounting rules to create subledger journal entries.

**source system**

An external system from a non-Oracle software provider, or internally created, that generates events which are to be accounted in the Oracle Fusion Accounting Hub.

**spot rate type**

Rate you enter to perform conversion based on this rate as of a specific date. This rate applies to the immediate delivery of a currency.

**subledger**

A low-level ledger that stores and manages the details that substantiate the monetary value stored in the general ledger. Oracle Fusion Receivables and Oracle Fusion Payables are examples of subledgers.

**subledger accounting options**

Defines how certain accounting processing should be done for transactions at the ledger and subledger application level. For example, the option to create a reversal entry includes displaying negative amounts or reversing the debit and credit sides.

**subledger journal entry**

A detailed journal entry generated for a transaction in a subledger application.

**subledger journal entry line**

An individual debit or credit line that is part of a subledger journal entry.

**subledger journal entry rule set**

A set of rules defining how to generate a complete journal entry for an accounting event.

**supporting reference**

Stores additional source information about a subledger journal entry either at the header or line level. They can be used to establish a subledger balance for a particular source value or combination of source values for a particular account.

**trading partner**

An external party, such as a supplier, in the Oracle B2B application for which electronic documents are sent or from which documents are received. A trading partner in Oracle B2B corresponds to a supplier site.

**transaction object**

Standardized data model containing transaction information used by the Create Accounting process to create subledger journal entries from accounting events.

**tree**

Information or data organized into a hierarchy with one or more root nodes connected to branches of nodes. A tree must have a structure where each node corresponds to data from one or more data sources.

**tree structure**

Characteristics applied to trees, such as what data to include or how the tree is versioned and accessed.

**tree version**

An instance of a tree. If a tree is associated with a reference data set, all versions belong to one set. Includes life cycle elements such as start and end date and a status indicator whether the tree is active or not.

**user rate type**

Rate you enter at journal entry time to convert foreign currency transactions to your functional currency.

**value set**

A set of valid values against which values entered by an end user are validated. The set may be tree structured (hierarchical).

**work area**

A set of tasks, reports, business intelligence, searches, and other content that a user needs to accomplish a business goal.