# Contents

## 1 Overview
- Product Management Offering: Overview ...................................................... 1-1
- Getting Started with an Implementation: Overview ........................................ 1-3
- Manage Application Implementation .............................................................. 1-4

## 2 Common Applications Configuration: Define Synchronization of Users and Roles from LDAP
- User and Role Synchronization: Explained .................................................. 2-1

## 3 Common Applications Configuration: Manage Inventory Organizations
- Inventory Organization: Critical Choices ..................................................... 3-1

## 4 Common Applications Configuration: Manage Units of Measure
- Units of Measure, Unit of Measure Classes, and Base Units of Measure: How They Fit Together ................................................................. 4-1
- Assigning Base Units of Measure to Unit of Measure Classes: Examples ........ 4-2
- Defining Unit of Measure Standard Conversions: Examples ....................... 4-2
- FAQs for Manage Units of Measure .............................................................. 4-3

## 5 Common Applications Configuration: Define Implementation Users
- Initial Security Administration: Critical Choices .......................................... 5-1
- Initial Security Administration: Worked Example ......................................... 5-3

## 6 Common Applications Configuration: Define Currencies and Currency Rates
- Manage Currencies ....................................................................................... 6-1
- Manage Conversion Rate Types ................................................................... 6-3
- Manage Daily Rates ..................................................................................... 6-7

## 7 Common Applications Configuration: Define Enterprise Structures
- Enterprise Structures: Overview ................................................................. 7-1
- Enterprise Structures Business Process Model: Explained ......................... 7-3
- Global Enterprise Configuration: Points to Consider ................................... 7-5
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling Your Enterprise Management Structure in Oracle Fusion: Example</td>
<td>7-5</td>
</tr>
<tr>
<td>Define Initial Configuration with the Enterprise Structures Configurator</td>
<td>7-8</td>
</tr>
<tr>
<td>Define Reference Data Sharing</td>
<td>7-33</td>
</tr>
<tr>
<td>Define Enterprise: Manage Enterprise HCM Information</td>
<td>7-40</td>
</tr>
<tr>
<td>Define Enterprise: Manage Locations</td>
<td>7-41</td>
</tr>
<tr>
<td>Define Geographies</td>
<td>7-43</td>
</tr>
<tr>
<td>Define Legal Entity</td>
<td>7-56</td>
</tr>
<tr>
<td>Define Chart of Accounts of Enterprise Structures: Manage Chart of Accounts</td>
<td>7-61</td>
</tr>
<tr>
<td>Define Chart of Accounts of Enterprise Structures: Manage Chart of Accounts Value Sets</td>
<td>7-76</td>
</tr>
<tr>
<td>Define Chart of Accounts of Enterprise Structures: Manage Account Hierarchies</td>
<td>7-80</td>
</tr>
<tr>
<td>Define Chart of Accounts of Enterprise Structures: Manage Accounting Calendars</td>
<td>7-98</td>
</tr>
<tr>
<td>Define Accounting Configurations of Enterprise Structures: Manage Primary or Ledgers and Subledgers: Explained</td>
<td>7-102</td>
</tr>
<tr>
<td>Define Accounting Configurations of Enterprise Structures: Manage Primary or Ledgers: Explained</td>
<td>7-102</td>
</tr>
<tr>
<td>Define Accounting Configurations of Enterprise Structures: Manage Primary or Ledgers: Points to Consider</td>
<td>7-103</td>
</tr>
<tr>
<td>Define Accounting Configurations of Enterprise Structures: Manage Primary or Financial Ledgers: How They Fit Together</td>
<td>7-107</td>
</tr>
<tr>
<td>Define Accounting Configurations of Enterprise Structures: Manage Primary or Creating Primary Ledgers: Example</td>
<td>7-109</td>
</tr>
<tr>
<td>Define Accounting Configurations of Enterprise Structures: Specify Ledger Options</td>
<td>7-110</td>
</tr>
<tr>
<td>Define Accounting Configurations of Enterprise Structures: Manage Reporting Currencies</td>
<td>7-114</td>
</tr>
<tr>
<td>Define Business Units: Manage Service Provider Relationships</td>
<td>7-117</td>
</tr>
<tr>
<td>Define Business Units: Specify Customer Contract Management Business Function Properties</td>
<td>7-120</td>
</tr>
<tr>
<td>Define Business Units: Assign Business Unit Business Function</td>
<td>7-125</td>
</tr>
<tr>
<td>Define Business Units: Manage Business Units</td>
<td>7-127</td>
</tr>
<tr>
<td>Define Facilities: Manage Facility Shifts, Workday Patterns, and Schedules</td>
<td>7-129</td>
</tr>
<tr>
<td>Define Facilities: Manage Inventory Organizations</td>
<td>7-132</td>
</tr>
<tr>
<td>Define Facilities: Manage Item Organizations</td>
<td>7-137</td>
</tr>
</tbody>
</table>

8 Common Applications Configuration: Define Security for Supply Chain Management

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Tasks: Highlights</td>
<td>8-1</td>
</tr>
<tr>
<td>Defining Security After Enterprise Setup: Points to Consider</td>
<td>8-4</td>
</tr>
<tr>
<td>Security Tasks and Oracle Fusion Applications: How They Fit Together</td>
<td>8-8</td>
</tr>
<tr>
<td>Security Tasks: Overview</td>
<td>8-11</td>
</tr>
<tr>
<td>Define Data Security</td>
<td>8-15</td>
</tr>
<tr>
<td>Define Users</td>
<td>8-27</td>
</tr>
</tbody>
</table>

9 Common Applications Configuration: Define Automated Governance, Risk, and Performance Controls

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segregation of Duties: Explained</td>
<td>9-1</td>
</tr>
<tr>
<td>Defining Segregation of Duties Policies: Points To Consider</td>
<td>9-4</td>
</tr>
<tr>
<td>Managing Segregation of Duties Risks and Violations: Critical Choices</td>
<td>9-6</td>
</tr>
<tr>
<td>Role Provisioning and Segregation of Duties: How They Work Together</td>
<td>9-8</td>
</tr>
</tbody>
</table>
15 Common Applications Configuration: Define Approval Management for Supply Chain Management

Approval Management: Highlights .................................................................................. 15-1

16 Product Information Management Configuration: Define Basic Items

Manage Product Value Sets ......................................................................................... 16-1
Manage Product Child Value Sets ............................................................................... 16-2
Manage Default Item Class ......................................................................................... 16-2
Manage Item Statuses ................................................................................................. 16-3
Manage Item Types ..................................................................................................... 16-3
Manage Cross Reference Types .................................................................................. 16-4
Manage Item Descriptive Flexfields ........................................................................... 16-4
Import Items ................................................................................................................ 16-6
Manage Related Item Subtypes ................................................................................... 16-7
FAQs for Define Basic Items ....................................................................................... 16-7

17 Cost Management Configuration: Define Cost Management

Manage Cost Organizations, Cost Organization Relationships, and Cost Books .......... 17-1
Manage Cost Analysis Groups .................................................................................... 17-6
Define Cost Policies and Cost Book Policies ............................................................... 17-7
Define Overheads ....................................................................................................... 17-18
Define Subledger Accounting Rules ........................................................................... 17-23

18 Cost Management Configuration: Define Receipt Accounting

Define Subledger Accounting Rules ............................................................................. 18-1

19 Define Transactional Business Intelligence Configuration

Define Transactional Business Intelligence Configuration: Highlights .................... 19-1
Access to Person Data ................................................................................................ 19-1

20 Define Extensions: Define Custom Enterprise Scheduler Jobs

Managing Job Definitions: Highlights ........................................................................ 20-1
Managing List of Values Sources: Highlights ............................................................ 20-2

21 Customization and Sandboxes

Customizing Pages Using Oracle Composer: Highlights ......................................... 21-1

22 Importing and Exporting Setup Data
Configuration Packages: Explained .................................................................22-1
Exporting and Importing Setup Data: Explained .......................................22-1
Moving Common Reference Objects .........................................................22-2
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.

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

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. You can use the Send Feedback to Oracle link in the footer of Oracle Fusion Applications Help.
Product Management Offering: Overview

By implementing the Product Management offering, your enterprise can configure how to create and manage items, catalogs, and related processes, as well as the structures to support receipt accounting for purchased items, cost accounting for financial reporting, and product profitability management. The Oracle Fusion Applications Product Information Management Implementation Guide covers implementation of item, catalog, and product management processes, while the Oracle Fusion Applications Cost Accounting and Receipt Accounting Implementation Guide covers implementation of cost management and receipt accounting processes.

Before you begin, use the Getting Started page in the Setup and Maintenance work area to access reports for each offering, including full lists of setup tasks, descriptions of the options and features that you can select when you configure the offering, and lists of business objects and enterprise applications that are associated with the offering.

The first implementation step is to configure the offerings in the Setup and Maintenance work area by selecting the offerings and options that you want to make available to implement. For the Product Management offering, you can select the following options:

- Product Management (this option also has associated features that you can select to configure)
- Product Model
- Basic Product Structures
- Basic Catalogs
- Product and Catalog Management
- Change Orders
- Product Rules
- Advanced Product Structures
- New Item Requests
- Product Hub
• Product Source Systems
• Advanced Catalogs
• Data Quality for Products
• Cost Management
• Receipt Accounting
• Product Information Management Business Intelligence Analytics
• Costing Business Intelligence Analytics
• Product Management Business Intelligence Analytics

Next, create one or more implementation projects for the offerings and options that you want to implement first, which generates task lists for each project. The application implementation manager can customize the task list and assign and track each task.

If you select all of the options, then the generated task list for this offering contains the following groups of tasks:

• Define Common Applications Configuration for Product Management
• Define Product Information Management Configuration
• Define Cost Management Configuration
• Define Transactional Business Intelligence Configuration
• Define Extensions for Product Management

**Define Common Applications Configuration for Product Management**

Use this task list to manage definitions that are used across offerings, which typically apply to multiple products and product families. These definitions include enterprise structures, workforce profiles, security, and approval rules, among others.

You can find other information that supports the common implementation tasks in the Oracle Fusion Applications Concepts Guide.

**Define Product Information Management Configuration**

Use this task list to configure items, product structures, catalogs, change orders, rules, new item requests, product source systems, and data quality.

**Define Cost Management Configuration**

Use this task list to configure cost accounting and receipt accounting processes, which enable organizations to calculate inventory item costs, maintain inventory valuation, generate accounting distributions for inventory transactions, and generate accounting distributions for receipt accruals.

Setting up cost accounting processes involves defining cost organization and cost accounting structures, cost books and policies, overhead absorption rules, analysis groups for reporting, and subledger and accounting rules that are used to post distributions to the general ledger for costing of inventory transactions.
Setting up receipt accounting processes involves defining subledger and accounting rules that are used to post distributions to the general ledger for receipt accruals.

**Define Transactional Business Intelligence Configuration**

Use this task list to configure Oracle Transactional Business Intelligence for ad hoc reporting, including managing the repository, connections, presentation catalog, and currency type display.

**Define Extensions for Product Management**

Use this task list to define extensions such as custom Oracle Enterprise Scheduler jobs.

You can also customize and extend applications using other tools. For more information, see the Oracle Fusion Applications Extensibility Guide.

## Getting Started with an Implementation: Overview

To start an Oracle Fusion Applications implementation, you must setup one or more initial users using the super user that was created during installation and provisioning of the Oracle Fusion Applications environment. Because Oracle Fusion Applications is secure as delivered, the process of enabling the necessary setup access for initial users requires several specialized steps when getting started with an implementation.

The following high level steps are required for starting an implementation.

1. As the Oracle Identity Management Administration user, provision the IT Security Manager job role with roles for user and role management.
   This step enables the super user account, which is provisioned with the IT Security Manager job role, to create implementation users.

2. As the Oracle Fusion Applications installation super user, perform the following tasks once you have selected an offering to implement and generated the setup tasks needed to implement the offering.
   b. Create an IT security manager user by using the Create Implementation Users task.
   c. Provision the IT security manager with the IT Security Manager role by using the Provision Roles to Implementation Users task.

3. As the newly created IT security manager user, sign in to Oracle Fusion Applications and set up at least one implementation user for setting up enterprise structures.
   a. Create an implementation user by using the Create Implementation Users task.
   b. Provision the implementation user with the Application Implementation Manager job role or the Application Implementation
Consultant job role by using the Provision Roles to Implementation Users task. The Application Implementation Consultant job role inherits from all product-specific application administrators and entitles the necessary View All access to all secured object.

c. Optionally, create a data role for an implementation user who needs only the limited access of a product-specific Application Administrator by using the Create Data Role for Implementation Users. Then assign the resulting data role to the implementation user by using the Provision Roles to Implementation Users task.

The figure shows the task flow from provisioning the IT Security Manager job role with the user and role management entitlement to creating and provisioning implementation users for enterprise setup.

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
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.
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.
User and Role Synchronization: Explained

Oracle Identity Management (OIM) maintains Lightweight Directory Access Protocol (LDAP) user accounts for users of Oracle Fusion applications. OIM also stores the definitions of abstract, job, and data roles and holds information about roles provisioned to users. During implementation, any existing information about users, roles, and roles provisioned to users must be copied from the LDAP directory to the Oracle Fusion Applications tables. Once the Oracle Fusion Applications tables are initialized with this information, it is maintained automatically. To perform the initialization, you run the process Retrieve Latest LDAP Changes.

Note

For security and audit best practice, implementation users have person records and appropriate role-based security access. So that appropriate roles can be assigned to implementation users, you must run the process Retrieve Latest LDAP Changes before you create implementation users.

Tip

During initial implementation, the installation super user performs the task Run User and Role Synchronization Process to run the Retrieve Latest LDAP Changes process.

The user name and password of the installation super user are created during installation provisioning of Oracle Fusion Applications. For details of the user name and password, contact your system administrator or the person who installed Oracle Fusion Applications.
Inventory Organization: Critical Choices

In Oracle Fusion, storage facilities, warehouses, and distribution centers are implemented as inventory organizations.

Inventory organizations are:

- Managed by a business unit, with the materials management business function enabled.
- Mapped to a legal entity and a primary ledger.

There are two types of inventory organizations:

- Manufacturing facilities
- Storage facilities

Storage and manufacturing facilities are related to other organizational entities through a business unit that stores, manufactures, and distributes goods through many factories, warehouses, and distribution centers. The material parameters are set for both the facilities, enabling movement of material in the organization. This business unit has the business function of Materials Management enabled. Oracle Fusion Applications allow many inventory organizations to be assigned to one business unit.

Note

Currently, Oracle Fusion Applications do not include manufacturing capabilities, so setup your manufacturing facilities outside of Oracle Fusion applications.

Distribution Center as an Inventory Organization

A distribution center can store inventory that is the responsibility of different business units. In this situation, assign an inventory organization to each business unit as a representation of the inventory in the distribution center. The multiple inventory organizations representing the inventory are defined with the same location to show that they are a part of the same distribution center.

In the following figure the two business units, Air Compressors and Air Transmission, share one distribution center in Atlanta. The two inventory
organizations, Air Compressors and Air Transmission represent the inventory for each business unit in the Atlanta distribution center and are both assigned the Atlanta location.

Legal Entities Own Inventory Organizations

A legal entity owns the inventory located in a storage or manufacturing facility. This ownership is assigned through the relationship of the inventory organization representing the inventory and the legal entity assigned to the inventory organization. The legal entity assigned to the inventory organization shares the same primary ledger as the inventory organization’s business unit.

The inventory is tracked in the inventory organization owned by the legal entity of which the business unit is part. All transactions are accounted for in the primary ledger of the legal entity that owns the inventory.

The figure below illustrates the inventory owned by InFusion Air Quality legal entity. The InFusion Air Quality legal entity is associated with the Air Compressors business unit, which is associated with the two Air Compressors inventory organizations. Therefore, InFusion Air Quality legal entity owns the entire inventory in both the Dallas and Atlanta locations.
Facility Schedules Are Associated with Inventory Organizations

A prerequisite to defining an inventory organization is to define a facility schedule. Oracle Fusion Applications allow you to associate an inventory organization with a schedule.

Facility schedules allow creating workday calendars for inventory organizations that are used in the Oracle Fusion Supply Chain Management product family. For example, use workday calendars in the scheduling of cycle counts and calculating transit time.
Define units of measure, unit of measure classes, and base units of measure for tracking, moving, storing, and counting items.

The Quantity unit of measure class contains the units of measure Box of 8, Box of 4, and Each. The unit of measure Each is assigned as the base unit of measure.

Unit of Measure Classes

Unit of measure classes represent groups of units of measure with similar characteristics such as area, weight, or volume.

Units of Measure

Units of measure are used by a variety of functions and transactions to express the quantity of items. Each unit of measure you define must belong to a unit of measure class.
Base Units of Measure

Each unit of measure class has a base unit of measure. The base unit of measure is used to perform conversions between units of measure in the class. For this reason, the base unit of measure should be representative of the other units of measure in the class, and should generally be one of the smaller units. For example, you could use CU (cubic feet) as the base unit of measure for a unit of measure class called Volume.

Assigning Base Units of Measure to Unit of Measure Classes: Examples

Each unit of measure class must have a base unit of measure.

Scenario

This table lists examples of unit of measure classes, the units of measure in each unit of measure class, and the unit of measure assigned as the base unit of measure for each unit of measure class. Note that each base unit of measure is the smallest unit of measure in its unit of measure class.

<table>
<thead>
<tr>
<th>Unit of Measure Class</th>
<th>Units of Measure</th>
<th>Base Unit of Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity</td>
<td>dozen</td>
<td>each</td>
</tr>
<tr>
<td></td>
<td>box</td>
<td></td>
</tr>
<tr>
<td></td>
<td>each</td>
<td></td>
</tr>
<tr>
<td>Weight</td>
<td>pound</td>
<td>gram</td>
</tr>
<tr>
<td></td>
<td>kilogram</td>
<td></td>
</tr>
<tr>
<td></td>
<td>gram</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>hour</td>
<td>second</td>
</tr>
<tr>
<td></td>
<td>minute</td>
<td></td>
</tr>
<tr>
<td></td>
<td>second</td>
<td></td>
</tr>
<tr>
<td>Volume</td>
<td>cubic feet</td>
<td>cubic inches</td>
</tr>
<tr>
<td></td>
<td>cubic centimeters</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cubic inches</td>
<td></td>
</tr>
</tbody>
</table>

Defining Unit of Measure Standard Conversions: Examples

A unit of measure standard conversion specifies the conversion factor by which the unit of measure is equivalent to the base unit of measure.

Scenario

This table lists examples of unit of measure classes, one unit of measure included in each class, the base unit of measure for the unit of measure class, and the conversion factor defined for the unit of measure.
<table>
<thead>
<tr>
<th>Unit of Measure Class</th>
<th>Unit of Measure</th>
<th>Base Unit of Measure</th>
<th>Conversion Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity</td>
<td>dozen</td>
<td>each</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(1 dozen = 12 each)</td>
</tr>
<tr>
<td>Weight</td>
<td>pound</td>
<td>gram</td>
<td>454</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(1 pound = 454 grams)</td>
</tr>
<tr>
<td>Time</td>
<td>minute</td>
<td>second</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(1 minute = 60 seconds)</td>
</tr>
</tbody>
</table>

FAQs for Manage Units of Measure

What's a unit of measure standard conversion?

A unit of measure standard conversion defines the conversion factor by which the unit of measure is equivalent to the base unit of measure that you defined for the unit of measure class. Defining a unit of measure standard conversion allows you to perform transactions in units other than the primary unit of measure of the item being transacted. The standard unit of measure conversion is used for an item if an item-specific unit of measure conversion has not been defined.

What's a UOM interclass conversion?

A UOM interclass conversion defines the conversion between the source base unit of measure ("From Base UOM") in one unit of measure class ("From Class") and the destination base unit of measure ("To Base UOM") in a different unit of measure class ("To Class").

For example, the item is gasoline. The From Base UOM (of the From Class called "volume") is liters. The To Base UOM (of the To Class called "quantity") is Barrels. The conversion is 158.76 liters (volume) to 1 barrel of oil (quantity).

What's a UOM intraclass conversion?

A UOM intraclass conversion specifies the conversion between a unit of measure (the "From UOM") and the base unit of measure of the same class.

For example, the item is soda pop. The unit of measure class is Quantity. The From UOM is Case (CS). The base unit of measure is Each (EA). The conversion is 24, to specify that 1 CS = 24 EA.
Initial Security Administration: Critical Choices

After installation and provisioning, and before setting up enterprise structures and implementing projects, you must establish required entitlement for the super user account and at least one implementation user to proceed with the implementation. Once initial enterprise structure setup is complete, additional users may be created through processes available in Human Capital Management (HCM).

Initial security administration consists of the following.

- Preparing the IT Security Manager job role
- Synchronizing users and roles from Lightweight Directory Access Protocol (LDAP) with HCM
- Creating implementation users
- Optionally creating data roles for implementation users
- Provisioning implementation users with roles

Once the first implementation project begins and the enterprise work structure is set up, use standard user and security management processes such as the Manage Users task to create and manage additional users. Do not use the Create Implementation Users task after your enterprise has been set up.

Preparing the IT Security Manager Job Role

Initially the super user is not provisioned to manage users and roles.

You must add the following Oracle Identity Management (OIM) roles to the IT Security Manager job role’s role hierarchy to enable the super user to create one or more initial implementation users.

- Identity User Administrators
- Role Administrators

Additionally, you must assign the Xellerate Users organization to the IT Security Manager role.
Synchronizing Users and Roles from LDAP

After configuring an offering and setting up the task lists for implementation, the Run User and Roles Synchronization Process task is available to the super user for synchronizing users and roles in the LDAP store with Oracle Fusion Human Capital Management (HCM).

Defining Initial Implementation Users

The super user is provisioned with roles that provide broad access to Oracle Fusion Middleware and Oracle Fusion Applications administration, and is not suitable as an implementation user in most enterprises. The super user should define at least one implementation user, which consists of creating the user account and provisioning it with at least the Application Implementation Consultant and Application Implementation Manager job roles.

As a security guideline, define an IT security manager user who in turn defines one or more implementation users to set up enterprise structures. The IT security manager users can provision the implementation user with the Application Implementation Consultant role, which entitles access to all enterprise structures. Or the IT security manager can create a data role that restricts access to enterprise structures of a specific product and provisioning that role.

Depending on the size of your implementation team, you may only need a single implementation user for security administration, implementation project management, enterprise structures setup, and application implementation. That single user must then be provisioned with all indicated roles, and therefore broad access.

Creating Implementation Users

The super user creates one or more implementation users by performing the Create Implementation Users task.

Note

This initial implementation user is a user account created in Oracle Identity Management only, specifically for setting up enterprise structures, and is not related to a real person or identity such as a user defined in HCM.

Creating Data Roles for Implementation Users

As an alternative to provisioning an implementation user with the Application Implementation Consultant role to access all enterprise structures, you may need implementation users with access restricted to enterprise structures for specific products. In this case, use the Create Data Roles for Implementation Users task to create a data role based on a job role with less broad access, such as the HCM Application Administrator job role.

Provisioning Roles to Implementation Users

After creating an implementation user, you must provision the user with one or more roles by performing the Provision Roles to Implementation Users task.
For example, assign a role to the implementation user that provides the access necessary for setting up the enterprise. Depending on need, provision to the implementation user the predefined Applications Implementation Consultant role or a product family-specific administrator data role, such as a data role based on the predefined Financials Applications Administrator.

---

**Caution**

The Application Implementation Consultant has broad access. It is a very useful role for experimentation or setting up a pilot environment, but may not be suitable for implementation users in a full implementation project.

---

**Initial Security Administration: Worked Example**

This example illustrates initial security administration after having installed and provisioned an Oracle Fusion Applications environment.

In Oracle Fusion Applications, you manage users and security through Oracle Fusion Human Capital Management (HCM) user management flows, which are included in each of the offering task lists. However, the HCM task flows require that enterprise structures have been set up, and yet to add users who can set up enterprise structures you need to have set up HCM. Therefore, you need to create one or more initial implementation users who are responsible for providing the following.

- Users and their applications security management
- Implementation project management
- Initial enterprise structures management

The following table summarizes key decisions for this scenario.

<table>
<thead>
<tr>
<th>Decision</th>
<th>In this Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>How to sign in to Oracle Fusion Applications for the first time</td>
<td>Use the super user account that was created when installing and provisioning Oracle Fusion Applications (for example, FAADMIN).</td>
</tr>
<tr>
<td>How to ensure that the roles and users in the Lightweight Directory Access Protocol (LDAP) store match what is available for selection when defining implementation users</td>
<td>Perform the Run User and Roles Synchronization Process task.</td>
</tr>
<tr>
<td>How to create a first implementation user</td>
<td>Prepare the IT Security Manager job role for user and role management so the super user and any other user provisioned with the IT Security Manager job role can manage users and roles.</td>
</tr>
<tr>
<td>How to establish security administration users</td>
<td>Define an IT security manager user provisioned with the IT Security Manager job role.</td>
</tr>
<tr>
<td>How to establish an implementation user with access to set up enterprise structures</td>
<td>Define an implementation user provisioned with the Application Implementation Consultant job role.</td>
</tr>
</tbody>
</table>

You create an initial implementation user by performing the following tasks.

1. The Oracle Identity Management System Administrator user provisions the IT Security Manager job role with roles for user and role management.
2. The Oracle Fusion Applications super user synchronizes LDAP users with HCM user management so that users can be provisioned with roles through HCM.

3. The Oracle Fusion Applications super user performs the Create Implementation Users task to create one or more IT security manager and administrator users provisioned with security administrative entitlement.

4. The IT Security Manager user signs in to Oracle Fusion Applications and performs the Create Implementation Users task to create implementation managers and users.

5. The IT Security Manager user provisions implementation users for enterprise structure setup.

---

**Note**

The following tasks assume that the super user has configured an offering and set up task lists. When not following a task flow within an activity, you can find tasks in Navigator > Tools > Setup and Maintenance > All Tasks. Search for the task and click its Go to Task icon in the search results.

---

**Preparing the IT Security Manager Role**

The super user that was created when installing and provisioning Oracle Fusion Applications (for example, FAADMIN) has all necessary access for implementing Oracle Fusion Applications and administering security. This access is provided by the following roles:

- Application Implementation Consultant
- IT Security Manager

Neither of these roles provides access needed for creating and managing Oracle Fusion Applications users. Therefore, you must add the following two OIM roles to the IT Security Manager role:

- Identity User Administrators
- Role Administrators

The following procedure is prerequisite to an IT security manager or administrator creating an initial one or more implementation users.

1. While signed into Oracle Identity Manager as the OIM System Administrator user, click the Administration link in the upper right of the Oracle Identity Manager.

   This accesses the Welcome to Identity Manager Delegated Administration menu.

2. In the Roles list of tasks, click Advanced Search - Roles. Search for the Identity Users Administrators role by entering the role name in Display Name and clicking Search.

   In the Search Results, click the role’s Display Name.

3. On the Hierarchy tab, select Inherits From and click Add.
4. In the Add Parent Role to: IDENTITY USER ADMINISTRATORS window, select the role category: Common - Job Roles and add the IT Security Manager.

   Click the arrow icon to show the list of available roles. Select IT Security Manager and move it to the Roles to Add list. Click Save.

5. Search for the Role Administrators role, and repeat steps 1 to 4 to add that role to the IT Security Manager role's role inheritance.

6. Assign the IT Security Manager role to the Xellerate Users organization.
   a. In the Welcome to Identity Manager Delegated Administration menu (see step 1, above), in the Organizations list of tasks, click Advanced Search - Organizations.
   b. Search for the Xellerate Users organization by entering Xellerate Users in Display Name and clicking Search.
   c. In the Search Results, click the organization's Display Name. The Xellerate Users page appears.
   d. Click the Administrative Roles link in the row of links above the Xellerate Users.
   e. In Filter By Role Name of the Details window, enter the following string:

      *IT_SECURITY_MANAGER*

      Click Find.
   f. Enable Read, Write, Delete, and Assign.
   g. Click Assign.
   h. Click Confirm.

   Synchronizing Users and Roles from LDAP

   Lightweight Directory Access Protocol (LDAP) must be synchronized with HCM user management so that users can be provisioned with roles through HCM.

   1. Sign in to Oracle Fusion Applications using the super user's user name (for example FAADMIN) and password.

      If you do not know the super user name and password, check with your system administrator or the person who installed Oracle Fusion Applications. For more information about account creation in Oracle Fusion Applications provisioning, see the Oracle Fusion Applications Installation Guide.

   2. Perform the Run User and Roles Synchronization Process task by clicking Submit in the Process Details page.

      The SyncRolesJob process takes some time to complete the first time it is run.

   3. Monitor completion of the SyncRolesJob process from Navigator > Tools > Scheduled Processes before continuing with creating implementation users.
Defining an IT Security Manager User

The super user has broad access to Oracle Fusion Middleware and Oracle Fusion Applications administration. Due to this broad access, your enterprise needs users dedicated to managing users and applications security, such as an IT security manager user.

1. While signed in as the Oracle Fusion Applications super user, access the Create Implementation Users task and create an IT security manager.

   The Oracle Identity Manager appears.

2. Click **Create User**.

   For details, see the Creating Users section in the Oracle Fusion Middleware User's Guide for Oracle Identity Manager.

3. Provide the following attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last name</td>
<td>&lt;any valid string&gt;</td>
<td>Smith</td>
</tr>
<tr>
<td>Organization</td>
<td>Xellerate Users</td>
<td>N/A</td>
</tr>
<tr>
<td>User type</td>
<td>Non Worker</td>
<td>N/A</td>
</tr>
<tr>
<td>User login</td>
<td>&lt;any valid string&gt;</td>
<td>IT_SECURITY_MANAGER</td>
</tr>
<tr>
<td>Login password</td>
<td>&lt;any valid string&gt;</td>
<td>SeKur1TyPa$$w0Rd</td>
</tr>
</tbody>
</table>

   **Note**

   In Oracle Fusion Applications, an implementation user is a user account created in OIM only, specifically for implementation tasks, and is not related to a real person or identity such as a user defined in HCM.

4. Click **Save**.

5. On the Roles tab in the IT_SECURITY_MANAGER user creation task flow, click **Assign**.

6. In the Add Role window, search for the IT Security Manager role and click **Add**.

Defining an Implementation User for Enterprise Structures Setup

1. Sign in to Oracle Fusion Applications using the IT security manager user’s name and password.

2. Create and provision an implementation user using the same task flow as for creating the IT security manager user in the previous section, except provision the following roles.

   - Application Implementation Manager
   - Application Implementation Consultant
Note

For an implementation to begin, at least one user must be provisioned with the Application Implementation Manager role, and another or the same user must be provisioned with the Application Implementation Consultant role. The Application Implementation Consultant has broad access to set up all enterprise structures.
Common Applications Configuration:
Define Currencies and Currency Rates

Manage 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.

**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 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.

<table>
<thead>
<tr>
<th>Action</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checked</td>
<td>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. 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.</td>
</tr>
<tr>
<td>Unchecked</td>
<td>General Ledger calculates the inverse rate but you can change the rate and update the daily rates table without the corresponding rate being updated.</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Currency Selected</th>
<th>Rate Type Selected</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAD</td>
<td>Corporate</td>
<td>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.</td>
</tr>
<tr>
<td>MXP</td>
<td>Spot</td>
<td>Entered a periodic type of transaction. Your company enters daily rates each day for MXP because this currency is unstable and fluctuates.</td>
</tr>
<tr>
<td>HKD</td>
<td>User</td>
<td>Entered a one time transaction. Your company does not maintain daily rates in HKD.</td>
</tr>
</tbody>
</table>

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?

<table>
<thead>
<tr>
<th>Rate Type</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot</td>
<td>For currencies with fluctuating conversion rates or when exact currency conversion is needed.</td>
</tr>
<tr>
<td>Corporate</td>
<td>For establishment of a standard rate across your organization for a stable currency.</td>
</tr>
<tr>
<td>User</td>
<td>For infrequent entries where your daily rates for the entered foreign currency are not set up.</td>
</tr>
</tbody>
</table>
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.

| Fixed | For rates where the conversion is constant between two currencies. |

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.
Common Applications Configuration: Define Enterprise Structures

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.
<table>
<thead>
<tr>
<th>BPM Activities</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define Enterprise</td>
<td>Define the enterprise to capture the name of the deploying enterprise and the location of the headquarters. 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.</td>
</tr>
<tr>
<td>Define Enterprise Structures</td>
<td>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.</td>
</tr>
<tr>
<td>Define Legal Jurisdictions and Authorities</td>
<td>Define information for governing bodies that operate within a jurisdiction.</td>
</tr>
<tr>
<td>Define Legal Entities</td>
<td>Define legal entities and legal reporting units for business activities handled by the Oracle Fusion Applications.</td>
</tr>
<tr>
<td>Define Business Units</td>
<td>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.</td>
</tr>
<tr>
<td>Define Financial Reporting Structures</td>
<td>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.</td>
</tr>
<tr>
<td>Define Chart of Accounts</td>
<td>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.</td>
</tr>
<tr>
<td>Define Ledgers</td>
<td>Define the primary accounting ledger and any secondary ledgers that provide an alternative accounting representation of the financial data.</td>
</tr>
<tr>
<td>Define Accounting Configurations</td>
<td>Define the accounting configuration that serves as a framework for how financial records are maintained for an organization.</td>
</tr>
<tr>
<td>Define Facilities</td>
<td>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.</td>
</tr>
<tr>
<td>Define Reference Data Sharing</td>
<td>Define how reference data in the applications is partitioned and shared.</td>
</tr>
</tbody>
</table>

**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
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 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.

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.
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 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, 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 then 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.

---

**Business Units: Explained**

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. A business unit can
process transactions on behalf of many legal entities. Normally, it will have a manager, strategic objectives, a level of autonomy, and responsibility for its profit and loss. Roll business units up into divisions if you structure your chart of accounts with this type of hierarchy. In Oracle Fusion Applications, you assign your business units to one primary ledger. For example, if a business unit is processing payables invoices they will need to post to a particular ledger. This assignment is mandatory for your business units with business functions that produce financial transactions.

In Oracle Fusion Applications, use business unit as a securing mechanism for transactions. For example, if you run your export business separately from your domestic sales business, secure the export business data to prevent access by the domestic sales employees. To accomplish this security, set up the export business and domestic sales business as two separate business units.

The Oracle Fusion Applications business unit model:

- Allows for flexible implementation
- Provides a consistent entity for controlling and reporting on transactions
- Anchors the sharing of sets of reference data across applications

Business units process transactions using reference data sets that reflect your business rules and policies and can differ from country to country. With Oracle Fusion Application functionality, you can choose to share reference data, such as payment terms and transaction types, across business units, or you can choose to have each business unit manage its own set depending on the level at which you wish to enforce common policies.

In countries where gapless and chronological sequencing of documents is required for subledger transactions, define your business units in alignment with your ledger definition, because the uniqueness of sequencing is only ensured within a ledger. In these cases, define a single ledger and assign one legal entity and business unit.

In summary, use business units in the following ways:

- Management reporting
- Processing of transactions
- Security of transactional data
- Reference data definition and sharing

**Brief Overview of Business Unit Security**

Business units are used by a number of Oracle Fusion Applications to implement data security. You assign data roles to your users to give them access to data in business units and permit them to perform specific functions on this data. When a business function is enabled for a business unit, the application can trigger the creation of data roles for this business unit based on the business function’s related job roles.

For example, if a payables invoicing business function is enabled, then it is clear that there are employees in this business unit that perform the function of payables invoicing, and need access to the payables invoicing functionality. Therefore, based on the correspondence between the business function and the job roles, appropriate data roles are generated automatically. Use Human Capital
Creating Business Units in the Enterprise Structures Configurator: Points to Consider

Business units are used within Oracle Fusion applications for management reporting, processing of transactions, and security of transactional data. Using the Enterprise Structures Configurator (ESC), you create business units for your enterprise either automatically or manually.

Automatically Creating Business Units

To create business units automatically, you must specify the level at which to create business units. Business units within your enterprise may be represented at the business function level, such as Sales, Consulting, Product Development, and so on, or they may be represented at a more detailed level, where a business unit exists for each combination of countries in which you operate and the functions in those countries.

You can automatically create business units at the following levels:

- Country
- Country and Division
- Country and business function
- Division
- Division and legal entity
- Division and business function
- Business function
- Legal entity
- Business function and legal entity

Select the option that best meets your business requirements, but consider the following:

- If you use Oracle Fusion Financials, the legal entity option is recommended because of the manner in which financial transactions are processed.
- The business unit level that you select determines how the application automatically creates reference data sets.

After you select a business unit level, the application generates a list of business units, and you select the ones you want the application to create. If you select a level that has two components, such as country and division, then the system displays a table listing both components, and you select the check boxes at the intersections of the components.

The business units listed by the application are suggestions only, and are meant to simplify the process to create business units. You are not required to select all of the business units suggested. When you navigate to the next page in the ESC guided flow, which is the Manage Business Units page, you cannot delete any of
the business units that were created automatically. You must return to the Create Business Units page and deselect any business units that you no longer want.

**Example: Selecting Business Unit Levels**

InFusion Corporation is using the Enterprise Structures Configurator to set up their enterprise structure. They have identified two divisions, one for Lighting, and one for Security. They operate in four countries: US, UK, Japan, and India, and they have created a legal entity for each of the countries. The sales and marketing functions are based in both India and Japan, while the US and the UK have only the sales function.

This figure illustrates InFusion Corporation’s enterprise structure.

The following table lists the options for business unit levels and the resulting business units that the application suggests for InFusion Corporation.

<table>
<thead>
<tr>
<th>Business Unit Level</th>
<th>Suggested Business Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
<td>• US</td>
</tr>
<tr>
<td></td>
<td>• UK</td>
</tr>
<tr>
<td></td>
<td>• Japan</td>
</tr>
<tr>
<td></td>
<td>• India</td>
</tr>
<tr>
<td>Country and Division</td>
<td>• InFusion Lighting: Japan</td>
</tr>
<tr>
<td></td>
<td>• InFusion Lighting: US</td>
</tr>
<tr>
<td></td>
<td>• Infusion Security: UK</td>
</tr>
<tr>
<td></td>
<td>• Infusion Security: India</td>
</tr>
<tr>
<td>Country and business function</td>
<td>• Sales: Japan</td>
</tr>
<tr>
<td></td>
<td>• Marketing: Japan</td>
</tr>
<tr>
<td></td>
<td>• Sales: US</td>
</tr>
<tr>
<td></td>
<td>• Sales: UK</td>
</tr>
<tr>
<td></td>
<td>• Marketing: India</td>
</tr>
<tr>
<td></td>
<td>• Sales: India</td>
</tr>
</tbody>
</table>
If none of the levels for creating business units meets your business needs, you can create business units manually, and you create them on the Manage Business Units page. If you create business units manually, then no reference data sets are created automatically. You must create them manually as well.

Reference Data Sets and Sharing Methods: Explained

Oracle Fusion Applications reference data sharing feature is also known as SetID. The reference data sharing functionality supports operations in multiple ledgers, business units, and warehouses, thereby reducing the administrative burden and decreasing the time needed to implement new business units. For example, you can share sales methods, transaction types, or payment terms across business units or selected other data across asset books, cost organizations, or project units.

The reference data sharing features use reference data sets to which reference data is assigned. The reference data sets group assigned reference data. The sets
can be understood as buckets of reference data assigned to multiple business units or other application components.

**Reference Data Sets**

You begin this part of your implementation by creating and assigning reference data to sets. Make changes carefully as changes to a particular set will affect all business units or application components using that set. You can assign a separate set to each business unit for the type of object that is being shared. For example, assign separate sets for payment terms, transaction types, and sales methods to your business units.

Your enterprise can decide that some aspects of corporate policy should affect all business units and leave other aspects to the discretion of the business unit manager. This allows your enterprise to balance autonomy and control for each business unit. For example, if your enterprise holds business unit managers accountable for their profit and loss, but manages working capital requirements at a corporate level, you can let managers define their own sales methods, but define payment terms centrally. In this case, each business unit would have its own reference data set for sales methods, and there would be one central reference data set for payment terms assigned to all business units.

The reference data sharing is especially valuable for lowering the cost of setting up new business units. For example, your enterprise operates in the hospitality industry. You are adding a new business unit to track your new spa services. The hospitality divisional reference data set can be assigned to the new business unit to quickly setup data for this entity component. You can establish other business unit reference data in a business unit specific reference data set as needed.

**Reference Data Sharing Methods**

There are variations in the methods used to share data in reference data sets across different types of objects. The following list identifies the methods:

- **Assignment to one set only, no common values allowed.** The simplest form of sharing reference data that allows assigning a reference data object instance to one and only one set. For example, Asset Prorate Conventions are defined and assigned to only one reference data set. This set can be shared across multiple asset books, but all the values are contained only in this one set.

- **Assignment to one set only, with common values.** The most commonly used method of sharing reference data that allows defining reference data object instance across all sets. For example, Receivables Transaction Types are assigned to a common set that is available to all the business units without the need to be explicitly assigned the transaction types to each business unit. In addition, you can assign a business unit specific set of transaction types. At transaction entry, the list of values for transaction types includes transaction types from the set assigned to the business unit, as well as transaction types assigned to the common set that is shared across all business units.

- **Assignment to multiple sets, no common values allowed.** The method of sharing reference data that allows a reference data object instance to be assigned to multiple sets. For instance, Payables Payment Terms use this method. It means that each payment term can be assigned to one or more than one set. For example, you assign the payment term Net 30 to several sets, but the payment term Net 15 is assigned to only your corporate business unit specific set. At transaction entry, the list of values...
for payment terms consists of only one set of data; the set that is assigned
to the transaction’s business unit.

Note: Oracle Fusion Applications contains a reference data set called Enterprise.
Define any reference data that affects your entire enterprise in this set.

**Business Units and Reference Data Sets: How They Work Together**

Reference data sharing is a feature within Oracle Fusion that enables you to group set-enabled reference data such as jobs or grades so that the data can be shared across different parts of the organization. Sets also enable you to filter reference data at the transaction level so that only data that has been assigned to certain sets is available to select. To filter reference data, Oracle Fusion Human Capital Management (HCM), applications use the business unit on the transaction. To set up reference data sharing in Oracle Fusion HCM, you create business units and sets, and then assign the sets to the business units.

**Common Set Versus Specific Sets**

Some reference data in your organization may be considered global, and should therefore be made available for use within the entire enterprise. You can assign this type of data to the Common Set, which is a predefined set. Regardless of the business unit on a transaction, reference data that has been assigned to the Common Set will always be available, in addition to the reference data that has been assigned to the set that corresponds to the business unit on the transaction.

Other types of reference data may be specific to certain business units, so you want to restrict the use of the data to those business units. In this case, you can create sets specifically for this type of data, and assign the sets to the business units.

**Business Unit Set Assignment**

When you assign reference data sets to business units, you assign a default reference data set that will be used for all reference data types for that business unit. You can override the set assignment for one or more data types.

**Example: Assigning Sets to Business Units**

InFusion Corporation has two divisions: Lighting and Security, and the divisions each have two locations. Each location has one or more business functions. The following figure illustrates the structure of InFusion Corporation.
When deciding how to create business units, InFusion decides to create them using the country and business function level. Therefore, they created the following business units:

- Sales_Japan
- Marketing_Japan
- Sales_US
- Sales_UK
- Marketing_India
- Sales_India

Because locations, departments, and grades are specific to each business unit, InFusion does not want to share these types of reference data across business units. They will create a reference data set for each business unit so that data of those types can be set up separately. Because the jobs in the Sales business function are the same across many locations, InFusion decides to create one additional set called Jobs and they will override the set assignment for the Jobs reference data group and assign it to the Jobs set. Based on these requirements, they create the following sets:

- Sales_Japan_Set
- Mktg_Japan_Set
- Sales_US_Set
- Sales_UK_Set
- Mktg_India_Set
- Sales_India_Set
- Grades_Set

InFusion assigns business units to sets as follows:

<table>
<thead>
<tr>
<th>Business Unit</th>
<th>Default Set Assignment</th>
<th>Set Assignment Overrides</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales_Japan</td>
<td>Sales_Japan_Set for grades, departments, and locations</td>
<td>Jobs set for jobs</td>
</tr>
<tr>
<td>Marketing_Japan</td>
<td>Mktg_Japan_Set for grades, departments, and locations</td>
<td>None</td>
</tr>
<tr>
<td>Sales_US</td>
<td>Sales_US_Set for grades, departments, and locations</td>
<td>Jobs set for jobs</td>
</tr>
<tr>
<td>Sales_UK</td>
<td>Sales_UK_Set for grades, departments, and locations</td>
<td>Jobs set for jobs</td>
</tr>
<tr>
<td>Marketing_India</td>
<td>Mktg_India_Set for grades, departments, and locations</td>
<td>None</td>
</tr>
<tr>
<td>Sales_India</td>
<td>Sales_India_Set for grades, departments, and locations</td>
<td>Jobs set for jobs</td>
</tr>
</tbody>
</table>

When setting up grades, departments, and locations for the business units, InFusion will assign the data to the default set for each business unit. When setting up jobs, they will assign the Jobs set and will assign the Common Set to any jobs that may be used throughout the entire organization.
When using grades, departments, and locations at the transaction level, users will be able to select data from the set that corresponds to the business unit that they enter on the transaction, and any data that was assigned to the Common Set. For example, for transactions for the Marketing_Japan business unit, grades, locations, and departments from the Mktg_Japan_Set will be available to select, as well as from the Common Set.

When using jobs at the transaction level, users will be able to select jobs from the Jobs set and from the Common Set when they enter one of the Sales business units on the transaction. For example, when a manager hires an employee for the Sales_India business unit, the list of jobs will be filtered to show jobs from the Jobs set and from the Common Set.

The following figure illustrates what sets of jobs can be accessed when a manager creates an assignment for a worker.

### Creating Reference Data Sets in the Enterprise Structures Configurator: Explained

If you created business units automatically, then the Enterprise Structures Configurator automatically creates reference data sets for you. The Enterprise Structures Configurator creates one reference data set for each business unit. You can add additional sets, but you cannot delete any of the sets that were created automatically.

A standard set called the Enterprise set is predefined.

#### Common Set

The common set is a predefined set that enables you to share reference data across business units. When you select set-enabled data at the transaction level, the list of values includes data in both the common set and the set associated with the data type for the business unit on the transaction. For example, when you create an assignment, the list of values for grades will include both grades in the common set and in the set that is assigned to grades for the business unit in which you creating the assignment.

#### Jobs and Positions: Critical Choices

Jobs and positions represent roles that enable you to distinguish between tasks and the individuals who perform those tasks. The key to whether to use jobs or
positions is how each is used. Positions offer a well-defined space independent of the person performing the job. Jobs are a space defined by the person. A job can be defined globally in the Common Set, whereas a position is defined within one business unit.

You can update the job and department of a position at any time. This is useful if you hire someone into a new role and want to transfer the position to another department.

During implementation, one of the earliest decisions you will make is whether to use jobs or a combination of jobs and positions. The determinants for this decision are:

- The primary industry of your enterprise
- How you manage your people

**Primary Industry of Your Enterprise**

Primary industries and how they usually set up their workforce are listed in the table below.

<table>
<thead>
<tr>
<th>Primary Industry</th>
<th>Workforce Setup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mining</td>
<td>Positions</td>
</tr>
<tr>
<td>Utilities</td>
<td>Positions</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>Positions</td>
</tr>
<tr>
<td>Retail Trade</td>
<td>Positions</td>
</tr>
<tr>
<td>Transportation and Warehousing</td>
<td>Positions</td>
</tr>
<tr>
<td>Educational Services</td>
<td>Positions</td>
</tr>
<tr>
<td>Public Transportation</td>
<td>Positions</td>
</tr>
<tr>
<td>Agriculture, Forestry, Fishing, and Hunting</td>
<td>Jobs</td>
</tr>
<tr>
<td>Construction</td>
<td>Jobs</td>
</tr>
<tr>
<td>Wholesale Trade</td>
<td>Jobs</td>
</tr>
<tr>
<td>Information</td>
<td>Jobs</td>
</tr>
<tr>
<td>Finance and Insurance</td>
<td>Jobs</td>
</tr>
<tr>
<td>Professional, Scientific, and Technical Services</td>
<td>Jobs</td>
</tr>
<tr>
<td>Management of Companies and Enterprises</td>
<td>Jobs</td>
</tr>
<tr>
<td>Administrative and Support and Waste Management and Remediation Services</td>
<td>Jobs</td>
</tr>
<tr>
<td>Arts, Entertainment, and Recreation</td>
<td>Jobs</td>
</tr>
<tr>
<td>Accommodation and Food Services</td>
<td>Jobs</td>
</tr>
<tr>
<td>Other Services (Except Public Administration)</td>
<td>Jobs</td>
</tr>
</tbody>
</table>

**Management of People**

The following table displays suggestions of whether to use jobs or a combination of jobs and positions based on your industry and how you manage your employees when there is turnover.
<table>
<thead>
<tr>
<th>Industry</th>
<th>We always replace employees by rehiring to same role</th>
<th>We replace the head count, but the manager can use the head count in a different job</th>
<th>We rehire to the same position, but the manager can request a reallocation of budget to a different post</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project (An industry that supports project-based forms of organization in which teams of specialists from both inside and outside the company report to project managers.)</td>
<td>Positions</td>
<td>Jobs</td>
<td>Jobs</td>
</tr>
<tr>
<td>Controlled (An industry that is highly structured in which all aspects of work and remuneration are well organized and regulated.)</td>
<td>Positions</td>
<td>Positions</td>
<td>Positions</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>Positions</td>
<td>Jobs</td>
<td>Positions</td>
</tr>
<tr>
<td>Retail</td>
<td>Positions</td>
<td>Jobs</td>
<td>Positions</td>
</tr>
<tr>
<td>Education</td>
<td>Positions</td>
<td>Jobs</td>
<td>Positions</td>
</tr>
<tr>
<td>Other</td>
<td>Positions</td>
<td>Jobs</td>
<td>Jobs</td>
</tr>
</tbody>
</table>

**Positions: Examples**

Positions are typically used by industries that use detailed approval rules, which perform detailed budgeting and maintain head counts, or have high turnover rates.

**Retail Industry**

ABC Corporation has high turnover. It loses approximately 5% of their cashiers monthly. The job of cashier includes three positions: front line cashier, service desk cashier, and layaway cashier. Each job is cross trained to take over another cashier position. When one cashier leaves from any of the positions, another existing cashier from the front line, service desk or layaway can assist where needed. But to ensure short lines and customer satisfaction, ABC must replace each cashier lost to turnover.

Since turnover is high in retail it is better for this industry to use positions. There is an automatic vacancy when an employee terminates employment. The position exists even when there are no holders. This is important if the person who leaves the company is a manager or supervisor with direct reports. All direct reports continue reporting to the position even if it is empty. You do not need to reassign these employees to another manager or supervisor; the replacement manager is assigned to the existing position.

Also, an advantage to using positions is that when you hire somebody new many of the attributes are defaulted in from the position. This speeds up the hiring process.

This figure illustrates the retail position setup.
The hospital has a structured head count and detailed budgeting. For example, a specific number of surgeons, nurses, and interns of various types are needed. These positions need to be filled in order for the hospital to run smoothly. Use jobs and positions if you need to apply detailed head count rules.

Health care is an industry that needs to regulate employment, roles, and compensation according to strict policies and procedures. Fixed roles tend to endure over time, surviving multiple incumbents. Industries that manage roles rather than individuals, where roles continue to exist after individuals leave, typically model the workforce using positions.

This figure illustrates the hospital position setup.

Jobs: Example

Jobs are typically used without positions by service industries where flexibility and organizational change are key features.
Software Industry

For example, XYZ Corporation has a director over the departments for developers, quality assurance, and technical writers. Recently, three developers have left the company. The director decides to redirect the head count to other areas. Instead of hiring all three back into development, one person is hired to each department, quality assurance, and technical writing.

In software industries, the organization is fluid. Using jobs gives an enterprise the flexibility to determine where to use head count, because the job only exists through the person performing it. In this example, when the three developers leave XYZ Corporation, their jobs no longer exist, therefore the corporation has the flexibility to move the headcount to other areas.

This figure illustrates the software industry job setup.

---

Job and Position Structures: Explained

Job and position structures identify the descriptive flexfield structure that enables you to specify additional attributes that you want to capture when you define jobs and positions. Job and position attributes provide further detail to make jobs and positions more specific. You also use attributes to define the structure of your jobs and positions. You can specify attributes at the enterprise level for jobs and positions, at the business unit level for positions, and at the reference data set level for jobs. Job and position structures are optional.

Enterprise-Level Job Attributes

When you define a job, you enter a value for the name of the job. To make job names more specific, set up attributes that enable you to identify additional details about the job, such as the nature of the work that is performed or the relative skill level required for the job. If these attributes apply to all jobs within your enterprise, set up enterprise-level job attributes. Standard capabilities mean that you can use the different segments of the name to identify common jobs or job holders for analysis or compensation, or for grouping records in reports, for example, to find all jobs of a specific job type. You should not use attributes with values that change regularly, for example, salary ranges or expense approval levels that change every year.

This figure illustrates how job type and job level provide further details for the HR Application Specialist job.
Common Applications Configuration: Define Enterprise Structures

Enterprise-Level Position Attributes

Position attributes at the enterprise level are similar to those for jobs. Each position that you define identifies a specific role in the enterprise, which you can manage independently of the person in the position, and it will belong to one specific department or organization. The name of each position must be unique. To simplify the process of managing unique names for positions, set up enterprise-level attributes to identify separate components of the position name. For example, you can set up an attribute for position title and one for position number. When defining the attributes that make up the structure of a position name you should also consider if any of your attributes are part of the definition of a common job type. Using job types for a position can help you manage common information that applies to many different positions. For example you can define a job type of Manager.Level 1 and use this for comparison of positions across departments or lines or business, or for setting common job requirements. You can then define multiple manager type positions in your HR department, each of which has responsibility for a different management function or group.

This figure illustrates how title and position number provide further details for the manager position.

Business Unit-Level Attributes for Positions

If you have information that you want to capture for positions that is specific to each business unit, then you can define attributes at the business unit level for positions. When you create positions, these attributes appear in addition to...
any enterprise-level attributes. For example, you may want to identify the sales region for all positions in the sales business unit. You can set up a text attribute called Sales Region and use it to enter the necessary information when creating positions for the sales business unit.

Reference Data Set-Level Attributes for Jobs

If you have information for jobs that applies to specific reference data sets, set up attributes for jobs at the reference data set level. When you create jobs, these attributes appear in addition to any enterprise-level attributes. For example, you may want to identify all information technology (IT) jobs within a specific set. You can set up a text attribute called Function and use it to enter IT in jobs that you create that perform an IT function within a specific set.

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.

What’s the default reference data set?

The reference data set that is assigned to a business unit for all reference data groups, such as grades, locations, departments, and jobs. You can override the default reference data set for any reference data group.

What happens if I override the set assignment?

For the selected business unit, you can override the default reference data set for one or more reference data groups. For example, assume you have three reference data groups: Vision 1 SET, Vision 2 SET, and Vision 3 SET, where Vision SET 1 is the default set for business unit United Kingdom Vision 1 BU. You can override the default so that grades are assigned to Vision 2 SET, departments are assigned to Vision 3 SET, and jobs are assigned to the default set, Vision 3 SET.
Define Reference Data Sharing

Reference Data Sharing: Explained

Reference data sharing facilitates sharing of configuration data such as jobs and payment terms, across organizational divisions or business units. You define reference data sets and determine how the data is shared or partitioned. Use reference data sets to reduce duplication and maintenance by sharing common data across business entities where appropriate. Depending on the requirement (specific or common), each business unit can maintain its data at a central location, using a set of values either specific to it or shared by other business units.

You can share reference data after it is filtered on the basis of sets. A common reference data set is available as the default set, which can be assigned to several business units sharing the same reference data. For commonly used data such as currencies, you can use the common reference data set and assign it to multiple business units in various countries that use the same currency. In cases where the default set cannot be assigned to an entity, you can create specific sets. The data set visible on the transactional page depends on the sharing method used to share reference data.

For example, XYZ Corporation uses the same grades throughout the entire organization. Instead of managers in different business units setting up the same grades, XYZ Corporation decides to create a set called Grades and assign the grades reference data group for all business units in the organization to the Grades set, so that the grades can be shared.

Note

For specific information on configuring reference data sharing for a particular object or product, refer to its product documentation.

Reference Data Sets: Explained

Reference data sets are logical groups of reference data that can be accessed by various transactional entities depending on the business context. Oracle Fusion Applications contains a common reference data set as well as an enterprise set that may be used as a default set. Depending on your business requirement you can create and maintain additional reference data sets, while continuing to use the common reference data set.

Consider the following scenario.

Your enterprise can decide that some aspects of corporate policy should affect all business units and leave other aspects to the discretion of the business unit manager. This allows your enterprise to balance autonomy and control for each business unit. For example, if your enterprise holds business unit managers accountable for their profit and loss, but manages working capital requirements
at a corporate level, you can let managers define their own sales methods, but define payment terms centrally. In this case, each business unit would have its own reference data set for sales methods, and there would be one central reference data set for payment terms assigned to all business units.

**Partitioning**

The partitioning of reference data and creation of data sets enable you to create reference entities across tables or lookup types, and share modular information and data processing options among business units. With the help of partitioning, you can choose to create separate sets and subsets for each business unit depending upon its business requirement, or create common sets or subsets to enable sharing reference data between several business units, without the need for duplicating the reference data. Partitioning provides you the flexibility to handle the reference data in a way appropriate to your business needs.

The following figure illustrates the reference data sharing method (assignment to one set only, with common values) where the user can access the data assigned to a specific set in a particular business unit, as well as access the data assigned to the common set.

**Reference Data Sets and Sharing Methods: Explained**

Oracle Fusion Applications reference data sharing feature is also known as SetID. The reference data sharing functionality supports operations in multiple ledgers, business units, and warehouses, thereby reducing the administrative burden and decreasing the time needed to implement new business units. For example, you can share sales methods, transaction types, or payment terms across business units or selected other data across asset books, cost organizations, or project units.

The reference data sharing features use reference data sets to which reference data is assigned. The reference data sets group assigned reference data. The sets can be understood as buckets of reference data assigned to multiple business units or other application components.
Reference Data Sets

You begin this part of your implementation by creating and assigning reference data to sets. Make changes carefully as changes to a particular set will affect all business units or application components using that set. You can assign a separate set to each business unit for the type of object that is being shared. For example, assign separate sets for payment terms, transaction types, and sales methods to your business units.

Your enterprise can decide that some aspects of corporate policy should affect all business units and leave other aspects to the discretion of the business unit manager. This allows your enterprise to balance autonomy and control for each business unit. For example, if your enterprise holds business unit managers accountable for their profit and loss, but manages working capital requirements at a corporate level, you can let managers define their own sales methods, but define payment terms centrally. In this case, each business unit would have its own reference data set for sales methods, and there would be one central reference data set for payment terms assigned to all business units.

The reference data sharing is especially valuable for lowering the cost of setting up new business units. For example, your enterprise operates in the hospitality industry. You are adding a new business unit to track your new spa services. The hospitality divisional reference data set can be assigned to the new business unit to quickly setup data for this entity component. You can establish other business unit reference data in a business unit specific reference data set as needed

Reference Data Sharing Methods

There are variations in the methods used to share data in reference data sets across different types of objects. The following list identifies the methods:

- Assignment to one set only, no common values allowed. The simplest form of sharing reference data that allows assigning a reference data object instance to one and only one set. For example, Asset Prorate Conventions are defined and assigned to only one reference data set. This set can be shared across multiple asset books, but all the values are contained only in this one set.

- Assignment to one set only, with common values. The most commonly used method of sharing reference data that allows defining reference data object instance across all sets. For example, Receivables Transaction Types are assigned to a common set that is available to all the business units without the need to be explicitly assigned the transaction types to each business unit. In addition, you can assign a business unit specific set of transaction types. At transaction entry, the list of values for transaction types includes transaction types from the set assigned to the business unit, as well as transaction types assigned to the common set that is shared across all business units.

- Assignment to multiple sets, no common values allowed. The method of sharing reference data that allows a reference data object instance to be assigned to multiple sets. For example, Payables Payment Terms use this method. It means that each payment term can be assigned to one or more than one set. For example, you assign the payment term Net 30 to several sets, but the payment term Net 15 is assigned to only your
corporate business unit specific set. At transaction entry, the list of values for payment terms consists of only one set of data; the set that is assigned to the transaction’s business unit.

Note: Oracle Fusion Applications contains a reference data set called Enterprise. Define any reference data that affects your entire enterprise in this set.

Assigning Reference Data Sets to Reference Objects: Points to Consider

You can assign the reference data sets to reference objects on the Manage Reference Data Set Assignments page. For multiple assignments, you can classify different types of reference data sets into groups and assign them to reference entity objects. The assignment takes into consideration the determinant type, determinant, and reference group, if any.

Determinant Types

The partitioned reference data is shared based on a business context setting called the determinant type. It is the point of reference used in the data assignment process. The following table lists the determinant types used in the reference data assignment.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset Book</td>
<td>Information about the acquisition, depreciation, and retirement of an asset that belongs to a ledger or a business unit.</td>
</tr>
<tr>
<td>Business Unit</td>
<td>The departments or organizations within an enterprise.</td>
</tr>
<tr>
<td>Cost Organization</td>
<td>The organization used for cost accounting and reporting on various inventory and cost centers within an enterprise.</td>
</tr>
<tr>
<td>Project Unit</td>
<td>A logical organization within an enterprise that is responsible for enforcing consistent project management practices.</td>
</tr>
<tr>
<td>Reference Data Set</td>
<td>References to other shared reference data sets.</td>
</tr>
</tbody>
</table>

Determinant

The determinant or determinant value is the value that corresponds to the selected determinant type. The determinant is one of the criteria for selecting the appropriate reference data set. For example, when managing set assignments for the set determinant type, Reference Data Set is the determinant type, and you would enter the corresponding set code value as the corresponding determinant value.

Reference Groups

A transactional entity may have multiple reference entities (generally considered to be setup data) that are treated in the same manner because of commonness in
implementing business policies and legal rules. Such reference entities in your application are grouped into logical units called reference groups, based on the functional area and the partitioning requirements that they have in common. For example, all tables and views that define Sales Order Type details might be part of the same reference group.

Note
The reference groups are predefined in the reference groups table and are available for selection and assignment.

Items and Supplier Site Reference Data Sharing: Explained

Some products required special logic for reference data sharing and have implemented their own domain specific ways for sharing data.

Items
If you share your items across warehouses or manufacturing facilities, you can access them through a common item master. Configure one or multiple item masters for your enterprise, based your enterprise structure. A single item master is recommended because it provides simpler and more efficient maintenance. However, in rare cases, it may be beneficial to keep multiple item masters. For example, if you acquire another enterprise and need to continue to operate your lines of business separately, maintaining a second item master might be the best decision.

Suppliers Sites
You can approve particular suppliers to supply specified commodities and authorize your business units to buy from those suppliers when the need arises. For example, you might be a household cleaning products manufacturer and need dyes, plastics, and perfumes to make your products. You purchase from a central supplier 70% of your perfume supplies with an additional supplier, in reserve, from whom you purchase the remaining 30%. At the same time, each of your business units purchases plastics and dyes from the same supplier, but from different local supplier sites to save transportation costs.

To implement business unit specific supplier sites, Oracle Fusion Procurement supports a method for defining suppliers sites as owned and managed by the business unit responsible for negotiating the supplier terms. Your other business units that have a service provider relationship defined with your procurement business unit, subscribe to the supplier sites using the supplier site assignments feature. In addition, Procurement allows sharing of the following procurement data objects across business units:

- Supplier qualification data, such as approved supplier lists
- Catalog content, such as agreements, smart forms, public shopping lists, and content zones
FAQs for Define Reference Data Sharing

What reference data objects can be shared across business units?

The following list contains the reference data objects for the Oracle Fusion Applications that can be shared across business units and the method in which the reference data for each is shared.

<table>
<thead>
<tr>
<th>Application Name</th>
<th>Reference Data Object</th>
<th>Method of Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trading Community Model</td>
<td>Customer Account Relationship</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Trading Community Model</td>
<td>Customer Account Site</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Trading Community Model</td>
<td>Sales Person</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Opportunity Management</td>
<td>Sales Method Group</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Work Management</td>
<td>Assessment Templates</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Enterprise Contracts</td>
<td>Contract Types</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Sales</td>
<td>Sales Method</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Common Components</td>
<td>Activity Templates</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Payables</td>
<td>Payment Terms</td>
<td>Assignment to multiple sets, no common values allowed</td>
</tr>
<tr>
<td>Receivables</td>
<td>Accounting Rules</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Receivables</td>
<td>Aging Buckets</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Receivables</td>
<td>Auto Cash Rules</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Receivables</td>
<td>Collectors</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Receivables</td>
<td>Memo Lines</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Receivables</td>
<td>Payment Terms</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Receivables</td>
<td>Remit To Address</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Receivables</td>
<td>Revenue Contingencies</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Receivables</td>
<td>Transaction Source</td>
<td>Assignment to one set only, with common values</td>
</tr>
</tbody>
</table>
### What reference data objects can be shared across asset books?

The following list contains the reference data objects for Oracle Fusion Assets that can be shared across asset books and the method in which the reference data for each is shared.

<table>
<thead>
<tr>
<th>Application Name</th>
<th>Reference Data Object</th>
<th>Method of Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assets</td>
<td>Bonus Rules</td>
<td>Assignment to one set only, no common values allowed</td>
</tr>
<tr>
<td>Assets</td>
<td>Depreciation Ceilings</td>
<td>Assignment to one set only, no common values allowed</td>
</tr>
<tr>
<td>Assets</td>
<td>Depreciation Methods</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Assets</td>
<td>Asset Descriptions</td>
<td>Assignment to one set only, no common values allowed</td>
</tr>
<tr>
<td>Assets</td>
<td>Property Types</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Assets</td>
<td>Prorate Conventions</td>
<td>Assignment to one set only, no common values allowed</td>
</tr>
</tbody>
</table>
Assets | Asset Queue Names | Assignment to one set only, with common values
--- | --- | ---
Assets | Retirement Types | Assignment to one set only, with common values
Assets | Unplanned Types | Assignment to one set only, with common values

What reference data objects can be shared across cost organizations?

The following list contains the reference data objects for Oracle Fusion Cost Management that can be shared across cost organizations and the method in which the reference data for each is shared.

<table>
<thead>
<tr>
<th>Application Name</th>
<th>Reference Data Object</th>
<th>Method of Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost Management</td>
<td>Cost Structure</td>
<td>Assignment to one set only, no common values allowed</td>
</tr>
</tbody>
</table>

What reference data objects can be shared across project units?

The following list contains the reference data objects for Oracle Fusion Project Foundation that can be shared across project units and the method in which the reference data for each is shared.

<table>
<thead>
<tr>
<th>Application Name</th>
<th>Reference Data Object</th>
<th>Method of Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Foundation</td>
<td>Project Definition</td>
<td>Assignment to multiple sets, no common values allowed</td>
</tr>
<tr>
<td>Project Foundation</td>
<td>Project Transaction Types</td>
<td>Assignment to multiple sets, no common values allowed</td>
</tr>
</tbody>
</table>

Define Enterprise: Manage Enterprise HCM Information

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 Enterprise: Manage Locations

Locations: Explained

A location identifies physical addresses of a workforce structure, such as a department or a job. You can also create locations to enter the addresses of external organizations that you want to maintain, such as employment agencies, tax authorities, and insurance or benefits carriers.

The locations that you create exist as separate structures that you can use for reporting purposes, and also in rules that determine employee eligibility for various types of compensation and benefits. You enter information about a location only once. Subsequently, when you set up other workforce structures you select the location from a list.

Location Sets

When you create a location, you must associate it with a set. Only those users who have access to the set’s business unit can access the location set and other associated workforce structure sets, such as those that contain departments and jobs.

You can also associate the location to the common set so that users across your enterprise can access the location irrespective of their business unit. When users search for locations, they can see the locations that they have access to along with the locations in the common set.
The following figure shows how locations sets restrict access to users.

Creating Multiple Locations Simultaneously

If you have a list of locations 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 location information to the spreadsheet, and then upload directly to your enterprise configuration. You can upload the spreadsheet multiple times to accommodate revisions.

FAQs for Manage Locations

Why can't I see my location in the search results?

You can search for approved locations only. Also, if you created a location in Oracle Fusion Trading Community Model, then you can't access that location from Oracle Fusion Global Human Resources. For use in Oracle Fusion HCM, you must recreate the location from the Manage Locations page.

What happens if I select a geographic hierarchy node when I'm creating or editing a location?

The calendar events that were created for the geographical node start to apply for the location and may impact the availability of worker assignments at that location. The geographical hierarchy nodes available for selection on the Locations page display from a predefined geographic hierarchy.

What happens if I select an inventory organization when I'm creating or editing a location?

The location is available for selection in purchase documents of that inventory organization in Oracle Fusion Inventory Management. If you don't select an inventory organization, then the location is available in purchase documents across all inventory organizations.
What happens if I inactivate a location?

Starting from the effective date that you entered, you can no longer associate the location with other workforce structures, assignments, or applications. If the location is already in use, it will continue to be available to the components that currently use it.

How can I associate a location with an inventory organization?

From the Manage Locations page in Oracle Fusion Global Human Resources.
To appear on the Create or Edit Location pages, your inventory organization must be effective on today’s date and must exist in the location set that you selected.

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.
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.

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.

**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:

<table>
<thead>
<tr>
<th>File-Based Import Entities</th>
<th>Interface Tables</th>
<th>Destination Tables</th>
<th>Application Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImpGeography</td>
<td>HZ_IMP_GEOGRAPHIES</td>
<td>HZ_GEOGRAPHIES</td>
<td>Geography</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HZ_GEOGRAPHY_IDEN</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HZ_GEOGRAPHY_TYPE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HZ_HIERARCHY_NODE</td>
<td></td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Decisions to Consider</th>
<th>In This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>What type of object are you importing?</td>
<td>Geography</td>
</tr>
<tr>
<td>What file type are you using for your source data?</td>
<td>Text file</td>
</tr>
<tr>
<td>Where are you uploading your source data file from?</td>
<td>Your desktop</td>
</tr>
<tr>
<td>What data type is your source data file?</td>
<td>Comma separated</td>
</tr>
<tr>
<td>Which fields are you importing into Oracle Fusion applications?</td>
<td>All, except for the RecordTypeCode field</td>
</tr>
<tr>
<td>When do you want to process the import?</td>
<td>Immediately</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Geography Level</th>
<th>Name</th>
<th>Source ID</th>
<th>Parent Source ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Country)</td>
<td>US</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2 (State)</td>
<td>CA</td>
<td>11</td>
<td>1</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Master Reference Geographies</td>
</tr>
<tr>
<td>Object</td>
<td>Geography</td>
</tr>
<tr>
<td>File Type</td>
<td>Text File</td>
</tr>
<tr>
<td>File Selection</td>
<td>Specific file</td>
</tr>
<tr>
<td>Upload From</td>
<td>Desktop</td>
</tr>
<tr>
<td>File Name</td>
<td>Choose relevant file from desktop</td>
</tr>
<tr>
<td>Data Type</td>
<td>Comma separated</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Column Header</th>
<th>Example Value</th>
<th>Ignore</th>
<th>Object</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary</td>
<td>Primary</td>
<td>United</td>
<td>Imp</td>
<td>Primary</td>
</tr>
<tr>
<td>Geography</td>
<td>Geography</td>
<td>States</td>
<td>Geography</td>
<td>Geography Name</td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td></td>
<td>Imp</td>
<td>Country</td>
</tr>
<tr>
<td>Country Code</td>
<td>US</td>
<td>No</td>
<td>Imp</td>
<td>Country Code</td>
</tr>
<tr>
<td>Record Type</td>
<td>0</td>
<td>Yes</td>
<td>Imp</td>
<td>Record Type</td>
</tr>
<tr>
<td>Code</td>
<td></td>
<td></td>
<td>Geography</td>
<td>Code</td>
</tr>
</tbody>
</table>
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.


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.
<table>
<thead>
<tr>
<th>Decisions to Consider</th>
<th>In This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy an existing country structure?</td>
<td>No, create a new country structure.</td>
</tr>
<tr>
<td>What is the structure of the geography types?</td>
<td>Create geography types with the following ranking</td>
</tr>
<tr>
<td></td>
<td>structure:</td>
</tr>
<tr>
<td></td>
<td>1. County</td>
</tr>
<tr>
<td></td>
<td>2. Post Town</td>
</tr>
<tr>
<td>What is the geography hierarchy?</td>
<td>Create the following hierarchy:</td>
</tr>
<tr>
<td></td>
<td>1. County of United Kingdom</td>
</tr>
<tr>
<td></td>
<td>2. County of Berkshire</td>
</tr>
<tr>
<td></td>
<td>3. Post Town of Reading</td>
</tr>
<tr>
<td>Which address style format will you use when</td>
<td>The default address style format, called the No</td>
</tr>
<tr>
<td>mapping geography validations?</td>
<td>Styles Format.</td>
</tr>
<tr>
<td>Are you using Oracle Fusion Tax for tax</td>
<td>No, do not select Tax Validation for the geography</td>
</tr>
<tr>
<td>purposes?</td>
<td>types.</td>
</tr>
</tbody>
</table>

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 Entity**

**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.

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.
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.

**Define Chart of Accounts of Enterprise Structures: Manage 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 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.
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.

**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 your 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.

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 Manager 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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment Code</td>
<td>INFUSION_AM_CO</td>
</tr>
<tr>
<td>Name</td>
<td>InFusion America Company</td>
</tr>
<tr>
<td>Description</td>
<td>InFusion America Inc. Company</td>
</tr>
<tr>
<td>Sequence Number</td>
<td>1</td>
</tr>
<tr>
<td>Prompt</td>
<td>Company</td>
</tr>
<tr>
<td>Short Prompt</td>
<td>CO</td>
</tr>
<tr>
<td>Display Width</td>
<td>2</td>
</tr>
<tr>
<td>Column Name</td>
<td>Segment1</td>
</tr>
<tr>
<td>Default Value Set Code</td>
<td>INFUSION_AM_COMPANY</td>
</tr>
</tbody>
</table>

   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 Manager 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**.

**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
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.

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.

### 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.

<table>
<thead>
<tr>
<th>Account</th>
<th>Debit</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company 1-Cost Center A-Advertising Expense Account</td>
<td>600</td>
<td></td>
</tr>
<tr>
<td>Company 2-Cost Center B-Advertising Expense Account</td>
<td></td>
<td>600</td>
</tr>
</tbody>
</table>
### 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.

---

<table>
<thead>
<tr>
<th>Account</th>
<th>Debit</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company 1-Cost Center A-Advertising Expense Account</td>
<td>600</td>
<td></td>
</tr>
<tr>
<td>Company 2-Cost Center B-Advertising Expense Account</td>
<td>600</td>
<td></td>
</tr>
<tr>
<td>Company 1-Cost Center A-Phone Expense Account</td>
<td>800</td>
<td></td>
</tr>
<tr>
<td>Company 2-Cost Center B-Phone Expense Account</td>
<td>800</td>
<td></td>
</tr>
<tr>
<td>Company 1-Cost Center A-Balancing Account</td>
<td>600</td>
<td></td>
</tr>
<tr>
<td>Company 2-Cost Center B-Balancing Account</td>
<td>800</td>
<td></td>
</tr>
<tr>
<td>Company 1-Cost Center A-Balancing Account</td>
<td>800</td>
<td></td>
</tr>
<tr>
<td>Company 2-Cost Center B-Balancing Account</td>
<td>800</td>
<td></td>
</tr>
</tbody>
</table>

---

**Define Chart of Accounts of Enterprise Structures: Manage Chart of Accounts Value Sets**

**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 setting 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 characteristic 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.

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.

<table>
<thead>
<tr>
<th>Segment Label Code</th>
<th>BI Object Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>FA_COST_CTR</td>
<td>Dim - Cost Center</td>
</tr>
<tr>
<td>GL_BALANCING</td>
<td>Dim - Balancing Segment</td>
</tr>
<tr>
<td>GL_ACCOUNT</td>
<td>Dim - Natural Account Segment</td>
</tr>
</tbody>
</table>

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.

**Define Chart of Accounts of Enterprise Structures: 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.

<table>
<thead>
<tr>
<th>Status</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft</td>
<td>Yet to be published or is in a modified state.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Validator</th>
<th>Description (what is checked)</th>
<th>Possible Cause for Validation Failure</th>
<th>Suggested Corrective Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Restrict By Set ID</strong></td>
<td>On the Manage Tree Structures: Specify Data Sources page, if the <strong>Set ID</strong> check box is selected to enable the <strong>Restrict Tree Node List of Values Based on</strong> 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.</td>
<td>Even when the check box is selected, one or more of its data source view objects does not contain a reference data set attribute.</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Row Flattened Table Name</strong></td>
<td>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 FND_TREE_NODE_RF or a custom table.</td>
<td>- The specified table does not exist in the database. - The specified table does not contain the same columns as the FND_TREE_NODE_RF table.</td>
<td>Correct the row flattened table definition.</td>
</tr>
</tbody>
</table>
| Available Label Data Sources | 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 **Labeling Scheme** 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 **None** from the **Labeling Scheme** list box. | • Any of the specified label data source view objects do not exist.  
• Any of the specified label data source view objects do not have primary keys.  
• 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.  
• Correct the specified label data source view object.  
• Correct the primary keys of the specified label data source view object.  
• Either correct the primary keys in the label data source view object to match the primary keys that were earlier registered in `FND_TS_DATA_SOURCE`, or correct the primary keys registered in that table to match the new view object definition. |
<table>
<thead>
<tr>
<th>Available Data Sources</th>
<th>Each data source view object specified for the tree structure must be accessible, and all its primary key attributes must be valid.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Any of the specified data source view objects do not exist.</td>
</tr>
<tr>
<td></td>
<td>• When a data source view object is initially defined, the database automatically registers the primary keys for the view object if the <strong>Use non-defined primary key columns</strong> 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.</td>
</tr>
<tr>
<td></td>
<td>• The <strong>Use non-defined primary key columns</strong> 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.</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td>• Correct the specified data source view object.</td>
</tr>
<tr>
<td></td>
<td>• Correct the duplicate column in the registered primary keys.</td>
</tr>
<tr>
<td></td>
<td>• Correct the primary keys of the specified data source view object.</td>
</tr>
<tr>
<td></td>
<td>• Correct any mismatch in data types.</td>
</tr>
<tr>
<td>Specifying Data Sources for Tree Structures: Points to Consider</td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
<td></td>
</tr>
</tbody>
</table>

- **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 `FND_TREE_NODE_CF` or a custom table.
  - The specified table does not exist in the database.
  - The specified table does not contain the same columns as the `FND_TREE_NODE_CF` table.
  - Correct the column flattened table definition.

- **Restrict by Date**
  - On the Manage Tree Structures: Specify Data Sources page, if the Date Range check box is selected to enable the Restrict Tree Node List of Values Based on 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 `FND_TREE_NODE` or a custom table.
  - No table is specified in the Tree Node Table field.
  - The specified table does not exist in the database.
  - The specified table does not contain the same columns as the `FND_TREE_NODE` table.
  - Correct the tree node table definition.

- **Allow Node Level Security**
  - If the Allow Node Level Security 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.
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.

<table>
<thead>
<tr>
<th>Labeling Scheme</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>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.</td>
</tr>
<tr>
<td>Group</td>
<td>Labels that you can arbitrarily assign to tree nodes.</td>
</tr>
<tr>
<td>Depth</td>
<td>Labels that are automatically assigned based on the depth of the tree node within the tree. No manual assignment is performed.</td>
</tr>
</tbody>
</table>

Note
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.

<table>
<thead>
<tr>
<th>Validator</th>
<th>Description (what is checked)</th>
<th>Possible Cause for Validation Failure</th>
<th>Suggested Corrective Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effective Date</td>
<td>The effective start and end dates of the tree version must be valid.</td>
<td>The effective end date is set to a value that is not greater than the effective start date.</td>
<td>Modify the effective start and end dates such that the effective start date is earlier than the effective end date.</td>
</tr>
<tr>
<td>Root Node</td>
<td>On the Manage Tree Structures: Specify Data Sources page, if the <strong>Allow Multiple Root Nodes</strong> check box for the <strong>Restrict Tree Node List of Values Based on</strong> 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.</td>
<td>Even if the check box is deselected, the tree version has multiple root nodes.</td>
<td>Modify the tree version such that there is exactly one root node.</td>
</tr>
<tr>
<td>Data Source Max Depth</td>
<td>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 <strong>Maximum Depth</strong> field is set to <strong>Unlimited</strong>.</td>
<td>The tree version has data at a depth greater than the specified depth limit on one or more data sources.</td>
<td>Modify the tree version such that all nodes are at a depth that complies with the data source depth limit.</td>
</tr>
<tr>
<td>Duplicate Node</td>
<td>On the Data Source dialog box, if the <strong>Allow Duplicates</strong> 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.</td>
<td>Even when the check box is deselected, the tree version contains duplicate nodes.</td>
<td>Remove any duplicate nodes from the tree version.</td>
</tr>
<tr>
<td>Available Node</td>
<td>All nodes in the tree version should be valid and available in the underlying data source.</td>
<td>• 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. • The tree version contains a tree reference node, which references another tree version that does not exist.</td>
<td>Remove any orphaned nodes from the tree version. Update tree reference nodes so that they reference existing tree versions.</td>
</tr>
<tr>
<td>Node Relationship</td>
<td>All nodes must adhere to the relationships mandated by the data sources registered in the tree structure.</td>
<td>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.</td>
<td>Modify the tree version such that the nodes adhere to the same parent-child relationships as the data sources.</td>
</tr>
<tr>
<td>SetID Restricted Node</td>
<td>On the Manage Tree Structures: Specify Data sources page, if the Set ID check box is selected to enable the <strong>Restrict Tree Node List of Values Based on</strong> 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.</td>
<td>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.</td>
<td>Modify the tree version such that all nodes in the tree have data sources with reference data set matching that of the tree.</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Label Enabled Node</td>
<td>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 <strong>Labeling Scheme</strong> list box, all nodes should have labels. This restriction does not apply when you select <strong>None</strong> from the <strong>Labeling Scheme</strong> list box.</td>
<td>The tree structure has a labeling scheme but the tree version has nodes without labels.</td>
<td>Assign a label to any node that does not have a label.</td>
</tr>
<tr>
<td>Date Restricted Node</td>
<td>On the Manage Tree Structures: Specify Data Sources page, if the <strong>Date Range</strong> check box is selected to enable the <strong>Restrict Tree Node List of Values Based on</strong> 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.</td>
<td>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).</td>
<td>Ensure that all nodes in the tree version have effective date range for the effective date range for the tree version.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Multiple Active Tree Version</td>
<td>On the Manage Tree Structures: Specify Definition page, if the <strong>Allow Multiple Active Tree Versions</strong> 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.</td>
<td>Even when the check box is not selected, there is more than one active tree version in the tree for the same date range.</td>
<td>Set no more than one tree version to Active within the same date range and set the others to inactive or draft status.</td>
</tr>
<tr>
<td>Range Based Node</td>
<td>On the Data Source dialog box, if the <strong>Allow Range Children</strong> 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.</td>
<td>Even when the check box is not selected, there are range-based nodes from a data source.</td>
<td>Ensure that any range nodes in your tree version are from a data source that allows range children.</td>
</tr>
<tr>
<td>Terminal Node</td>
<td>On the Data Source dialog box, if the Allow Use as Leaves 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.</td>
<td>Even when the check box is not selected, values from a data source are added as leaf nodes (terminal nodes).</td>
<td>Modify the tree version such that all terminal nodes are from data sources for which this check box is selected.</td>
</tr>
<tr>
<td>Usage Limit</td>
<td>On the Data Source dialog box, if the Use All Values option is selected to set the Usage Limit for the data source, every value in the data source must appear as a node in the tree. This restriction does not apply if None option is selected.</td>
<td>Even if the Use All Values option is selected, there are values in the data source that are not in the tree version.</td>
<td>For each data source value that is not yet available, add nodes to the tree version.</td>
</tr>
</tbody>
</table>

**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
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:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATUS_CODE</td>
<td>Enter the value NEW to indicate that you are bringing new segment value data.</td>
</tr>
<tr>
<td>VALUE_SET_CODE</td>
<td>Enter the value set code for the segment values.</td>
</tr>
<tr>
<td>VALUE</td>
<td>Enter the segment value.</td>
</tr>
<tr>
<td>SUMMARY_FLAG</td>
<td>Select N if the segment value is a child value or Y if the segment value is a parent value.</td>
</tr>
<tr>
<td>ENABLED_FLAG</td>
<td>Select Y to enable the segment value. Enter N to disable the segment value.</td>
</tr>
<tr>
<td>ACCOUNT_TYPE:</td>
<td>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.</td>
</tr>
<tr>
<td>ALLOW_POSTING_FLAG</td>
<td>Select Y if posting is allowed for this segment value. Select N if posting is not allowed.</td>
</tr>
<tr>
<td>OBJECT_VERSION_NUMBER</td>
<td>Enter default value of 1.</td>
</tr>
</tbody>
</table>

You can enter values for the following optional columns:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>START_DATE_ACTIVE</td>
<td>Enter the start date of the segment value</td>
</tr>
<tr>
<td>END_DATE_ACTIVE</td>
<td>Enter the end date of the segment value</td>
</tr>
<tr>
<td>THIRD_PARTY_CTRL_ACCOUNT</td>
<td>Enter the third party control account value. Valid values are: CUSTOMER, SUPPLIER, R for Restrict Manual Journals, Y, and N.</td>
</tr>
</tbody>
</table>
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:

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

Define Chart of Accounts of Enterprise Structures: Manage
Accounting 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.

<table>
<thead>
<tr>
<th>Validation Performed</th>
<th>Example of Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique period number</td>
<td>2 assigned for two periods</td>
</tr>
<tr>
<td>Unique period name</td>
<td>Jan-11 entered twice</td>
</tr>
<tr>
<td>Period number beyond the maximum number of periods per year</td>
<td>13 for a 12 period calendar with no adjusting periods</td>
</tr>
<tr>
<td>Entered period name contains spaces</td>
<td>Jan 11</td>
</tr>
<tr>
<td>Single or double quotes in the period name</td>
<td>Jan '11</td>
</tr>
<tr>
<td>Nonadjusting periods with overlapping dates</td>
<td>01-Jan-2011 to 31-Jan-2011 and 30-Jan-2011 to 28-Feb-2011</td>
</tr>
<tr>
<td>Period date gaps</td>
<td>01-Jan-2011 to 28-Jan-2011 and 31-Jan-2011 to 28-Feb-2011</td>
</tr>
<tr>
<td>Missing period numbers</td>
<td>Periods 1 through 6 defined for a twelve month calendar</td>
</tr>
<tr>
<td>Period number gaps</td>
<td>1, 3, 5</td>
</tr>
</tbody>
</table>
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 cannot 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 Accounting Configurations of Enterprise Structures: Manage Primary or Secondary Ledgers

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.

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 of 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.

_Authored by Oracle_
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 task generally done after the accounting setups 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:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>InFusion America</td>
</tr>
<tr>
<td>Description</td>
<td>InFusion America primary ledger for recording transactions.</td>
</tr>
<tr>
<td>Chart of Accounts</td>
<td>InFusion America Chart of Accounts</td>
</tr>
<tr>
<td>Accounting Calendar</td>
<td>Standard Monthly</td>
</tr>
<tr>
<td>Currency</td>
<td>USD</td>
</tr>
<tr>
<td>Accounting Method</td>
<td>Standard Accrual</td>
</tr>
</tbody>
</table>

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.

**Define Accounting Configurations of Enterprise Structures: Specify Ledger Options**

**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.
Both primary and secondary ledgers are created in the same way and use the same user interface to enable their specific ledger options.

**Note**

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.

Completing the Period Close Region Options

1. Enter your Retained Earnings Account: 101-00-31330000-0000-0000-0000-0000. 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: 101-00-31350000-0000-0000-0000-0000. 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.
<table>
<thead>
<tr>
<th>Option</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Suspense</td>
<td>General Ledger</td>
</tr>
<tr>
<td>Default Expense Account</td>
<td>101-00-98199999-0000-0000-0000-0000-0000</td>
</tr>
<tr>
<td>Rounding Account</td>
<td>101-10-98189999-0000-0000-0000-0000-0000</td>
</tr>
<tr>
<td>Entered Currency Balancing Account</td>
<td>101-10-98179999-0000-0000-0000-0000-0000</td>
</tr>
<tr>
<td>Balancing Threshold Percent</td>
<td>10</td>
</tr>
</tbody>
</table>

2. Click all the following Entry options listed in the table.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable journal approval</td>
<td>Click to enable journal approval functionality. Approval rules must be created in the Oracle Fusion Approvals Management (AMX).</td>
</tr>
<tr>
<td>Notify when prior period journal</td>
<td>Notify the user when a prior period date is selected on a journal entry.</td>
</tr>
<tr>
<td>Allow mixed and statistical journals</td>
<td>Enter both monetary and statistical amounts on the same line in a journal entry.</td>
</tr>
<tr>
<td>Validate reference date</td>
<td>Requires a reference date in an open or future enterable period.</td>
</tr>
</tbody>
</table>

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
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 used 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.

### Define Accounting Configurations of Enterprise Structures: 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.

<table>
<thead>
<tr>
<th>Type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period Opening Tasks</td>
<td>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.</td>
</tr>
<tr>
<td>Daily Tasks</td>
<td>Enter the daily conversion rates to convert your journals to each of your reporting currencies.</td>
</tr>
<tr>
<td>Period Closing Tasks</td>
<td>• Finish entering all regular and adjusting journals for the period in your ledger.</td>
</tr>
<tr>
<td></td>
<td>• Post all unposted journals in your ledger if not already done in the previous step.</td>
</tr>
<tr>
<td></td>
<td>• Post all unposted journals in your reporting currencies if not already done in the previous step.</td>
</tr>
<tr>
<td></td>
<td>• Run Revaluation in both your ledger and reporting currencies. Post the resulting revaluation batches in each ledger.</td>
</tr>
<tr>
<td></td>
<td>• As needed, translate balances in your ledger.</td>
</tr>
<tr>
<td></td>
<td>• Generate needed reports from both your ledger and reporting currencies.</td>
</tr>
<tr>
<td></td>
<td>• Close your accounting period in both your ledger and reporting currencies.</td>
</tr>
</tbody>
</table>

**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.

Define Business Units: Manage Service Provider Relationships

Shared Service Centers: Explained

Oracle Fusion Applications allows defining relationships between business units to outline which business unit provides services to the other business units.

Service Provider Model

In Oracle Fusion Applications V1.0, the service provider model centralizes only the procurement business function. Your business units that have the requisitioning business function enabled can define relationships with business units that have the procurement business function enabled. These service provider business units will process requisitions and negotiate supplier terms for their client business units.

This functionality is used to frame service level agreements and drive security. The definition of service provider relationships provides you with a clear record of how the operations of your business are centralized. For other centralized processing, business unit security is used (known in Oracle EBS as Multi-Org Access Control). This means that users who work in a shared service center have the ability to get access and process transactions on behalf of many business units.

Shared Service Center: Points to Consider

Oracle Fusion applications supports shared service centers in two ways. First, with business unit security, which allows your shared service centers personnel to process transactions for other business units called clients. This was the foundation of Multi Org Access Control in the Oracle E-Business Suite.
Second, the service provider model expands on this capability to allow a business unit and its personnel in a shared service center to work on transactions of the client business units. It is possible to view the clients of a service provider business unit, and to view service providers of a client business unit.

Your shared service centers provide services to your client business units that can be part of other legal entities. In such cases, your cross charges and recoveries are in the form of receivables invoices, and not merely allocations within your general ledger, thereby providing internal controls and preventing inappropriate processing.

For example, in traditional local operations, an invoice of one business unit cannot be paid by a payment from another business unit. In contrast, in your shared service center environment, processes allowing one business unit to perform services for others, such as paying an invoice, are allowed and completed with the appropriate intercompany accounting. Shared service centers provide your users with access to the data of different business units and can comply with different local requirements.

Security

The setup of business units provides you with a powerful security construct by creating relationships between the functions your users can perform and the data they can process. This security model is appropriate in a business environment where local business units are solely responsible for managing all aspects of the finance and administration functions.

In Oracle Fusion applications, the business functions your business unit performs are evident in the user interface for setting up business units. To accommodate shared services, use business unit security to expand the relationship between functions and data. A user can have access to many business units. This is the core of your shared service architecture.

For example, you take orders in many business units each representing different registered legal entities. Your orders are segregated by business unit. However, all of these orders are managed from a shared service order desk in an outsourcing environment by your users who have access to multiple business units.

Benefits

In summary, large, medium, and small enterprises benefit from implementing share service centers. Examples of functional areas where shared service centers are generally implemented include procurement, disbursement, collections, order management, and human resources. The advantages of deploying these shared service centers are the following:

- Reduce and consolidate the number of control points and variations in processes, mitigating the risk of error.
- Increase corporate compliance to local and international requirements, providing more efficient reporting.
- Implement standard business practices, ensuring consistency across the entire enterprise and conformity to corporate objectives.
• Establish global processes and accessibility to data, improving managerial reporting and analysis.
• Provide quick and efficient incorporation of new business units, decreasing startup costs.
• Establish the right balance of centralized and decentralized functions, improving decision making.
• Automate self-service processes, reducing administrative costs.
• Permit business units to concentrate on their core competencies, improving overall corporate profits.

Service Provider Model: Explained

In Oracle Fusion applications, the service provider model defines relationships between business units for a specific business function, identifying one business in the relationship as a service provider of the business function, and the other business unit as its client.

Procurement Example

The Oracle Fusion Procurement product family has taken advantage of the service provide model by defining outsourcing of the procurement business function. Define your business units with requisitioning and payables invoicing business functions as clients of your business unit with the procurement business function. Your business unit responsible for the procurement business function will take care of supplier negotiations, supplier site maintenance, and purchase order processing on behalf of your client business units. Subscribe your client business units to the supplier sites maintained by the service providers, using a new procurement feature for supplier site assignment.

In the InFusion example below, business unit four (BU4) serves as a service provider to the other three business units (BU1, BU2, and BU3.) BU4 provides the corporate administration, procurement, and human resources (HR) business functions, thus providing cost savings and other benefits to the entire InFusion enterprise.
Define Business Units: Specify Customer Contract Management Business Function Properties

Customer Contracts Business Unit Setup: Explained

Using the Specify Customer Contract Management Business Function Properties task, available by navigating to Setup and Maintenance work area and searching on the task name, you can specify a wide variety of business function settings for customer contracts in a specific business unit. The selections you make for these business functions impact how Oracle Fusion Enterprise Contracts behaves during contract authoring.

Using the Specify Customer Contract Management Business Function Properties task, manage these business function properties:

- Enable related accounts
- Set currency conversion details
- Manage project billing options
- Set up the Contract Terms Library

The setup options available for the Contract Terms Library are applicable to both customer and supplier contracts, and are described in the business unit setup topic for the Contract Terms Library. That topic is available as a related link to this topic.

Enabling Related Customer Accounts

Contract authors can specify bill-to, ship-to, and other accounts for the parties in a contract. Enable the related customer accounts option if you want accounts previously specified as related to the contract party to be available for selection.

Managing Currency Conversion Options

If your organization plans to transact project-related business in multiple currencies, then select the multicurrency option. This allows a contract author to override a contract's currency, which defaults from the ledger currency of the business unit. It also enables the contract author to specify currency conversion attributes to use when converting from the bill transaction currency to the contract currency and from the invoice currency to the ledger currency.

In the Bill Transaction Currency to Contract Currency region, enter currency conversion details that will normally be used, by all contracts owned by this business unit, to convert transaction amounts in the bill transaction currency to the contract currency. Newly created contracts contain the default currency conversion values, but you can override the values on any contract, if needed.

In the Invoice Currency to Ledger Currency region:

- Enter invoice transaction conversion details if the invoice and ledger currencies can be different.
• Enter revenue transaction conversion details if the revenue and ledger currencies can be different for as-incurred and rate-based revenue.

Managing Project Billing Options

The options available for selection in the Project Billing region control the behavior of project invoicing and revenue recognition for contracts with project-based work.

Project billing can behave differently for external contracts (customer billing) or intercompany and interproject contracts (internal billing).

Set these options, which apply to all contracts:

• Select the Transfer Revenue to General Ledger option if you want to create revenue accounting events and entries, and transfer revenue journals to the general ledger. If this option is not selected, then revenue can still be generated, but will not be transferred to the general ledger.

• Indicate if a reason is required for credit memos that are applied to invoices.

There are two sets of the following options, one for customer billing and a second for internal billing:

• Select an invoice numbering method, either Manual or Automatic. The invoice numbering method is the method that Oracle Fusion Receivables uses to number its invoices, upon release of draft invoices from Project Billing.

  • If the invoice numbering method is Manual, then select an invoice number type, which sets the type of Receivables invoice numbers that are allowed. Valid values are Alphanumeric and Numeric.

  • If the invoice numbering method is Automatic, then enter the next invoice number to use when generating Receivables invoice numbers.

• Select the Receivables batch source to use when transferring invoices to Receivables.

Set this option only for customer billing:

• Indicate if you want contract authors to manually enter the Receivables transaction type on the customer contracts they create.

Contract Terms Library Business Unit Setup: Explained

You can specify a wide variety of Contract Terms Library settings for either customer or supplier contracts within each business unit, by using either the Specify Customer Contract Management Business Function Properties or the Specify Supplier Contract Management Business Function Properties tasks. These tasks are available by navigating to the Setup and Maintenance work area and searching on the task name.

For the Contract Terms Library in each business unit, you can:

• Enable clause and template adoption.
• Set the clause numbering method.

• Enable the Contract Expert feature.

• Specify the layout for printed clauses and contract deviation reports.

**Enabling Clause Adoption**

If you plan to use clause adoption in your implementation, then set up the following:

- **Specify a global business unit**

  You must designate one of the business units in your organization as the global business unit by selecting the Global Business Unit option. This makes it possible for the other local business units to adopt and use approved content from that global business unit. If the Global Business Unit option is not available for the business unit you are setting up, this means that you already designated another business unit as global.

- **Enable automatic adoption**

  If you are implementing the adoption feature, then you can have all the global clauses in the global business unit automatically approved and available for use in the local business by selecting the Autoadopt Global Clauses option. If you do not select this option, the employee designated as the Contract Terms Library Administrator must approve all global clauses before they can be adopted and used in the local business unit. This option is available only for local business units.

- **Specify the administrator who approves clauses available for adoption**

  You must designate an employee as the Contract Terms Library administrator if you are using adoption. If you do not enable automatic adoption, then the administrator must adopt individual clauses or localize them for use in the local business unit. The administrator can also copy over any contract terms templates created in the global business unit. The clauses and contract terms templates available for adoption are listed in the administrator’s Terms Library work area.

**Setting Clause Numbering Options**

You can set up automatic clause numbering for the clauses in the business unit by selecting Automatic in the Clause Numbering field and entering a Document Sequence Category you previously set up in the Clause Sequence Category field. If clause numbering is manual, contract terms library administrators must enter unique clause numbers each time they create a clause.

You can choose to display the clause number in front of the clause title in contracts by selecting the Display Clause Number in Clause Title option.

**Enabling Contract Expert**

You must select the Enable Contract Expert option to be able to use the Contract Expert feature in a business unit. This setting takes precedence over enabling Contract Expert for individual contract terms templates.
Specifying the Printed Clause and Deviations Report Layouts

For each business unit, you can specify the Oracle BI Publisher RTF file that serves as the layout for:

- The printed contract terms

  Enter the RTF file you want used for formatting the printed clauses in the Clause Layout Template field.

- The contract deviations report

  The RTF file you select as the Deviations Layout Template determines the appearance of the contract deviations report PDF. This PDF is attached to the approval notification sent to contract approvers.

Define Business Units: Specify Supplier Contract Management Business Function Properties

Supplier Contracts Business Unit Setup: Explained

Using the Specify Supplier Contract Management Business Function Properties task, available by selecting Setup and Maintenance from the Tools menu and searching on the task name, you can specify a variety of business function settings for supplier contracts in a specific business unit.

The selections you make for these business functions impact how the Contract Terms Library behaves during supplier contract authoring.

Managing Contract Terms Library Setup Options

The setup options available for the Contract Terms Library are applicable to both customer and supplier contracts, and are described in the business unit setup topic for the Contract Terms Library. That topic is available as a related link to this topic.

Contract Terms Library Business Unit Setup: Explained

You can specify a wide variety of Contract Terms Library settings for either customer or supplier contracts within each business unit, by using either the Specify Customer Contract Management Business Function Properties or the Specify Supplier Contract Management Business Function Properties tasks. These tasks are available by navigating to the Setup and Maintenance work area and searching on the task name.

For the Contract Terms Library in each business unit, you can:

- Enable clause and template adoption.
- Set the clause numbering method.
- Enable the Contract Expert feature.
Specify the layout for printed clauses and contract deviation reports.

Enabling Clause Adoption

If you plan to use clause adoption in your implementation, then set up the following:

- Specify a global business unit
  
  You must designate one of the business units in your organization as the global business unit by selecting the **Global Business Unit** option. This makes it possible for the other local business units to adopt and use approved content from that global business unit. If the **Global Business Unit** option is not available for the business unit you are setting up, this means that you already designated another business unit as global.

- Enable automatic adoption
  
  If you are implementing the adoption feature, then you can have all the global clauses in the global business unit automatically approved and available for use in the local business by selecting the **Autoadopt Global Clauses** option. If you do not select this option, the employee designated as the Contract Terms Library Administrator must approve all global clauses before they can be adopted and used in the local business unit. This option is available only for local business units.

- Specify the administrator who approves clauses available for adoption
  
  You must designate an employee as the Contract Terms Library administrator if you are using adoption. If you do not enable automatic adoption, then the administrator must adopt individual clauses or localize them for use in the local business unit. The administrator can also copy over any contract terms templates created in the global business unit. The clauses and contract terms templates available for adoption are listed in the administrator’s Terms Library work area.

Setting Clause Numbering Options

You can set up automatic clause numbering for the clauses in the business unit by selecting Automatic in the **Clause Numbering** field and entering a Document Sequence Category you previously set up in the **Clause Sequence Category** field. If clause numbering is manual, contract terms library administrators must enter unique clause numbers each time they create a clause.

You can choose to display the clause number in front of the clause title in contracts by selecting the **Display Clause Number in Clause Title** option.

Enabling Contract Expert

You must select the **Enable Contract Expert** option to be able to use the Contract Expert feature in a business unit. This setting takes precedence over enabling Contract Expert for individual contract terms templates.

Specifying the Printed Clause and Deviations Report Layouts

For each business unit, you can specify the Oracle BI Publisher RTF file that serves as the layout for:
• The printed contract terms
  Enter the RTF file you want used for formatting the printed clauses in the Clause Layout Template field.

• The contract deviations report
  The RTF file you select as the Deviations Layout Template determines the appearance of the contract deviations report PDF. This PDF is attached to the approval notification sent to contract approvers.

Define Business Units: Assign Business Unit Business Function

Business Functions: Explained

A business unit can perform many business functions in Oracle Fusion Applications. Prior to Oracle Fusion Applications, operating units in Oracle E-Business Suite were assumed to perform all business functions, while in Oracle PeopleSoft, each business unit had one specific business function. Oracle Fusion Applications blends these two models and allows defining business units with one or many business functions.

Business Functions

A business function represents a business process, or an activity that can be performed by people working within a business unit and describes how a business unit is used. The following business functions exist in Oracle Fusion applications:

• Billing and revenue management
• Collections management
• Customer contract management
• Customer payments
• Expense management
• Incentive compensation
• Marketing
• Materials management
• Inventory management
• Order fulfillment orchestration
• Payables invoicing
• Payables payments
• Procurement
• Procurement contract management
• Project accounting
• Receiving
• Requisitioning
• Sales

Although there is no relationship implemented in Oracle Fusion Applications, a business function logically indicates a presence of a department in the business unit with people performing tasks associated with these business functions. A business unit can have many departments performing various business functions. Optionally, you can define a hierarchy of divisions, business units, and departments as a tree over HCM organization units to represent your enterprise structure.

Note
This hierarchy definition is not required in the setup of your applications, but is a recommended best practice.

Your enterprise procedures can require a manager of a business unit to have responsibility for their profit and loss statement. However, there will be cases where a business unit is performing only general and administrative functions, in which case your manager’s financial goals are limited to cost containment or recovering of service costs. For example, if a shared service center at the corporate office provides services for more commercially-oriented business units, it does not show a profit and therefore, only tracks its costs.

In other cases, where your managers have a responsibility for the assets of the business unit, a balance sheet can be produced. The recommended best practice to produce a balance sheet, is to setup the business unit as a balancing segment in the chart of accounts. The business unit balancing segment can roll up to divisions or other entities to represent your enterprise structure.

When a business function produces financial transactions, a business unit must be assigned to 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.

The following business functions generate financial transactions and will require a primary ledger and a default legal entity:

• Billing and revenue management
• Collections management
• Customer payments
• Expense management
• Materials management
• Payables invoicing
• Project accounting
• Receiving
• Requisitioning

**Business Unit Hierarchy: Example**

For example, your InFusion America Company provides:

- Air quality monitoring systems through your division InFusion Air Systems
- Customer financing through your division InFusion Financial Services

The InFusion Air Systems division further segments your business into the System Components and Installation Services subdivisions. Your subdivisions are divided by business units:

- System Components by products: Air Compressors and Air Transmission
- Installation Services by services: Electrical and Mechanical

Oracle Fusion applications facilitates independent balance sheet rollups for legal and management reporting by offering up to three balancing segments. Hierarchies created using the management segment can provide the divisional results. For example, it is possible to define management segment values to correspond to business units, and arrange them in a hierarchy where the higher nodes correspond to divisions and subdivisions, as in the Infusion US Division example above.

**Define Business Units: Manage Business Units**

**Business Units: Explained**

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. A business unit can process transactions on behalf of many legal entities. Normally, it will have a manager, strategic objectives, a level of autonomy, and responsibility for its profit
and loss. Roll business units up into divisions if you structure your chart of accounts with this type of hierarchy. In Oracle Fusion Applications, you assign your business units to one primary ledger. For example, if a business unit is processing payables invoices they will need to post to a particular ledger. This assignment is mandatory for your business units with business functions that produce financial transactions.

In Oracle Fusion Applications, use business unit as a securing mechanism for transactions. For example, if you run your export business separately from your domestic sales business, secure the export business data to prevent access by the domestic sales employees. To accomplish this security, set up the export business and domestic sales business as two separate business units.

The Oracle Fusion Applications business unit model:

- Allows for flexible implementation
- Provides a consistent entity for controlling and reporting on transactions
- Anchors the sharing of sets of reference data across applications

Business units process transactions using reference data sets that reflect your business rules and policies and can differ from country to country. With Oracle Fusion Application functionality, you can choose to share reference data, such as payment terms and transaction types, across business units, or you can choose to have each business unit manage its own set depending on the level at which you wish to enforce common policies.

In countries where gapless and chronological sequencing of documents is required for subledger transactions, define your business units in alignment with your ledger definition, because the uniqueness of sequencing is only ensured within a ledger. In these cases, define a single ledger and assign one legal entity and business unit.

In summary, use business units in the following ways:

- Management reporting
- Processing of transactions
- Security of transactional data
- Reference data definition and sharing

**Brief Overview of Business Unit Security**

Business units are used by a number of Oracle Fusion Applications to implement data security. You assign data roles to your users to give them access to data in business units and permit them to perform specific functions on this data. When a business function is enabled for a business unit, the application can trigger the creation of data roles for this business unit based on the business function's related job roles.

For example, if a payables invoicing business function is enabled, then it is clear that there are employees in this business unit that perform the function of payables invoicing, and need access to the payables invoicing functionality. Therefore, based on the correspondence between the business function and the job roles, appropriate data roles are generated automatically. Use Human Capital
Define Facilities: Manage Facility Shifts, Workday Patterns, and Schedules

Schedule Components: How They Fit Together

Schedules are comprised of workday patterns and exceptions. Workday patterns are comprised of shifts. You can also create exceptions, nonworking days, to the schedules.

Begin by creating shifts and then assigning those shifts to workday patterns. Next, create a schedule that is a collection of workday patterns and any exception dates.

A shift is a period of time, typically expressed in hours, and it can be defined by a start time and an end time, or a duration. A shift can be for a work period or a off period. You can create time, duration, and elapsed shifts.
**Workday Pattern**

A workday pattern is a collection of shifts for a specific number of days. You can create time, duration, and elapsed workday patterns.

**Exception**

An exception is a record of a date that overrides the availability of a resource to which a schedule has been assigned. For example, a resource is assigned a schedule that includes December 25 as a working day. An exception can be created for December 25 and applied to that schedule to override resource availability for that date. Exceptions can also be for a date time period such as 9 a.m. to 11 a.m. on December 25th.

**Schedule**

A schedule is defined by a start date, an end date, and a sequence of workday patterns to be followed between those dates. A schedule can also contain exception dates that override the availability of resources to which the schedule is assigned. Quarter types such as 4-4-5, 4-5-4 are supported.

---

**Managing Shifts: Examples**

A shift is a period of time, typically expressed in hours, that is used to build workday patterns. Workday patterns are used to build schedules. There are multiple types of shifts you can create. The following scenarios illustrate each type.

**Managing Time Shifts**

Next month you are adding a second shift for your manufacturing operations. This new shift will start right after your regular first shift. You can create a time shift that starts at 4:00 p.m. and ends at 12:00 a.m. There are restrictions in updating existing shifts and patterns. Shifts and patterns cannot be updated if the change affects a schedule, that is they are associated to a schedule. If a shift is created but not assigned to a pattern (or assigned to a pattern but the pattern is not assigned to a schedule) it can be updated. If a pattern is created and not assigned to a schedule it can be updated.

**Managing Time Shifts with Punch Details**

Your division has decided that the employees in the office must clock in and out for lunch starting next week. All employees will take the same lunch hour. Add punch shift details to the existing shift so that employees punch in at 8:00 a.m.; they punch out for lunch from 11:30 a.m. to 12:30 p.m.; they punch back in at 12:30 p.m.; and they punch out for the day at 5:00 p.m.

**Managing Time Shifts with Flexible Details**

Jorge Sanchez is a contractor who is starting work in your department next week. His hours will be flexible, so you need to create a new time shift with flexible details that he can use to record his time. He will have a flexible start time from 7:00 a.m. to 9:00 a.m. and a flexible end time from 4:00 p.m. to 6:00 p.m. His core work hours will be from 9:00 a.m. to 4:00 p.m.
Managing Duration Shifts

One of the divisions in your organization does not use fixed start and end times for its daily shifts; the division only records the total duration of the shift and indicates if resources are available or not during that time. All of the employees in the division are available for 24 hours straight, and then they are not available for the next 24 hours. You should create a duration shift that indicates that resources are available for 24 hours, and create a second duration shift that indicates that resources are not available for 24 hours.

Managing Elapsed Shifts

The employees in the Human Resources department all work 8 hours a day, but the start and end times vary by employee. Some employees start at early as 6:00 a.m., while others don’t start until 9:00 a.m. Create an elapsed shift with a duration of 8 hours, where all employees are assumed to be available for the number of hours in the shift at any time during the day.

Managing Workday Patterns: Examples

A workday pattern is a collection of shifts for a specific number of days. There are multiple types of workday patterns you can create. The following scenarios illustrate each type.

Managing Time Workday Patterns

Your department works a Monday through Friday workweek with 8 hour shifts each day. Time patterns always have time shifts. That is, the shift will have start time and end time. You can create a time workday pattern with a length of 7 days and details of an 8 hour time shift for days 1 through 5. Days 6 and 7 are considered nonworking days.

Managing Duration Workday Patterns

A new group of employees starts next month, and each employee will work a schedule where he or she is available for 10 hours, and then not available for the next 16 hours, and then available for 10 hours again, and so on. This pattern starts on midnight of the first day of the next month. Create a duration workday pattern with a 10-hour available duration shift, followed by a 16-hour not available duration shift. Do not specify the pattern length or start and end days, and the pattern will repeat for the length of the schedule to which it is associated.

Managing Elapsed Workday Patterns

In the summer, several divisions in your organization work only 4 hours on Fridays. They work extended hours on Wednesdays and Thursdays to cover the 4 hours they will not work on Fridays. Create an elapsed workday pattern with a length of 7 days. Days 1 and 2 will have an 8-hour shift assigned, while days 3 and 4 will have a 10-hour shift assigned. Finally, day 5 will have a 4-hour shift assigned. As in the time workday pattern, days 6 and 7 are considered nonworking days.
Define Facilities: Manage Inventory Organizations

Inventory Organizations: Explained

An inventory organization is a logical or physical entity in the enterprise that is used to store definitions of items or store and transact items.

You select the following usages in the inventory organization's properties:

- Item management
- Item and inventory management

Item Management

Inventory organizations used for item management, which are the same as item organizations, store only definitions of items. Use inventory organizations for item management when the storage or movement of inventory does not need to be physically or financially tracked. For example, in a retail implementation you can create an inventory organization for item management to store the names of items that are listed by and sold through each retail outlet, while a different system tracks physical inventory and transactions. If it is necessary in the future, you can change an inventory organization's usage from item management to item and inventory management in the inventory organization's properties.

Item and Inventory Management

Inventory organizations used for item and inventory management store and transact items, in addition to item definitions. An inventory organization used for item and inventory management is associated with one business unit, one legal entity, and one primary ledger. Use inventory organizations for item and inventory management when the storage or movement of inventory needs to be physically and financially tracked. Inventory organizations used for item and inventory management can represent facilities such as manufacturing centers, warehouses, or distribution centers. You cannot change an inventory organization's use from item and inventory management to item management.

Inventory Organization: Critical Choices

In Oracle Fusion, storage facilities, warehouses, and distribution centers are implemented as inventory organizations.

Inventory organizations are:

- Managed by a business unit, with the materials management business function enabled.
- Mapped to a legal entity and a primary ledger.

There are two types of inventory organizations:

- Manufacturing facilities
- Storage facilities
Storage and manufacturing facilities are related to other organizational entities through a business unit that stores, manufactures, and distributes goods through many factories, warehouses, and distribution centers. The material parameters are set for both the facilities, enabling movement of material in the organization. This business unit has the business function of Materials Management enabled. Oracle Fusion Applications allow many inventory organizations to be assigned to one business unit.

Note

Currently, Oracle Fusion Applications do not include manufacturing capabilities, so setup your manufacturing facilities outside of Oracle Fusion applications.

Distribution Center as an Inventory Organization

A distribution center can store inventory that is the responsibility of different business units. In this situation, assign an inventory organization to each business unit as a representation of the inventory in the distribution center. The multiple inventory organizations representing the inventory are defined with the same location to show that they are a part of the same distribution center.

In the following figure the two business units, Air Compressors and Air Transmission, share one distribution center in Atlanta. The two inventory organizations, Air Compressors and Air Transmission represent the inventory for each business unit in the Atlanta distribution center and are both assigned the Atlanta location.

Legal Entities Own Inventory Organizations

A legal entity owns the inventory located in a storage or manufacturing facility. This ownership is assigned through the relationship of the inventory organization representing the inventory and the legal entity assigned to the
The inventory organization. The legal entity assigned to the inventory organization shares the same primary ledger as the inventory organization’s business unit.

The inventory is tracked in the inventory organization owned by the legal entity of which the business unit is part. All transactions are accounted for in the primary ledger of the legal entity that owns the inventory.

The figure below illustrates the inventory owned by InFusion Air Quality legal entity. The InFusion Air Quality legal entity is associated with the Air Compressors business unit, which is associated with the two Air Compressors inventory organizations. Therefore, InFusion Air Quality legal entity owns the entire inventory in both the Dallas and Atlanta locations.

Facility Schedules Are Associated with Inventory Organizations

A prerequisite to defining an inventory organization is to define a facility schedule. Oracle Fusion Applications allow you to associate an inventory organization with a schedule.

Facility schedules allow creating workday calendars for inventory organizations that are used in the Oracle Fusion Supply Chain Management product family. For example, use workday calendars in the scheduling of cycle counts and calculating transit time.

Inventory Organization Prerequisites: Points to Consider

You can create a new inventory organization, or select an existing organization to define as an inventory organization.

Before creating inventory organizations:

- Set up inventory organization dependencies
- Plan inventory organization parameters

Setting Up Inventory Organization Dependencies

When you create an inventory organization, you must associate it to dependencies, such as business units and legal entities. For this reason, create these dependencies before creating an inventory organization.
Planning Inventory Organization Parameters

Before creating an inventory organization, plan the inventory organization’s parameters.

Consider the following when planning to configure an inventory organization’s parameters:

- Which schedule to use
- Which inventory organization to serve as the item master organization
- Whether to configure locator control and if so, the level at which to enforce the locator control
- How you want to configure movement request settings such as pick slip batch size and replenishment movement request grouping

Consider the size of your operation, your usage of subinventories, and the type of labor or equipment required when considering whether you want to use organization- or subinventory-level replenishment movement request grouping.

- How you want to configure lot, serial, and packing unit generation settings

To make appropriate choices for these settings, you should be familiar with:

- Your company’s guidelines for creating lot names, serial numbers, and packing unit numbers
- Whether your company requires you to assign the same lot number to multiple items in the same organization, or a specific lot number to only one item in the same organization
- Whether your company requires you to place purchase order or shipping order material under lot control
- How you want to configure item sourcing details, such as the picking rule to use, and whether to specify the inventory organization as a logistics services organization

Rounding the Reorder Quantity: How It Affects Min-Max Planning Reorder Calculations

When you specify to round reorder quantities, min-max planning reorders for item subinventories are automatically rounded up or down.

Settings That Affect Rounding the Reorder Quantity

Reorder quantities for an item subinventory are calculated based on:
- The setting that you select for the **Round Order Quantity** parameter on the Manage Inventory Organization Parameters page, General tab, of the inventory organization containing the item subinventory
- The value that you specify for the **Fixed Lot Multiple** text box on the Add Item to Subinventory window

### How Rounding the Reorder Quantity Affects Min-Max Planning Reorder Quantity Calculations

If you enable rounding the reorder quantity for the inventory organization, and specify the fixed lot multiple for the item subinventory, the reorder quantity is rounded up. If you disable rounding the reorder quantity for the inventory organization, and specify the fixed lot multiple for the item subinventory, the reorder quantity is rounded down.

---

**Note**

To round reorder quantities, you must specify a fixed lot multiple.

---

**Example: Rounding the Reorder Quantity**

Assume that the reorder quantity is 24. If you enable rounding the reorder quantity and specify 10 for the fixed lot multiple, the reorder quantity is rounded up to 30. If you disable rounding the reorder quantity and keep the fixed lot multiple at 10, the reorder quantity is rounded down to 20.

### Selecting Lot Number Uniqueness Control: Critical Choices

Select one of the following lot number uniqueness control options to apply to the items in your inventory organization:

- No uniqueness control
- Across items

**No Uniqueness Control**

You can assign the same lot number to multiple items in the same inventory organization and across inventory organizations. The following table provides an example of how lot numbers are generated when uniqueness control is not applied, both within and across inventory organizations.

<table>
<thead>
<tr>
<th>Within Inventory Organization</th>
<th>Across Inventory Organizations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item AS100 (printer) / Lot LN100</td>
<td>Item AS100 (printer) / Lot LN100</td>
</tr>
<tr>
<td>Item AS101 (laptop computer) / Lot LN100</td>
<td>Item AS101 (laptop computer) / Lot LN100</td>
</tr>
</tbody>
</table>

**Across Items**

You can only assign a unique lot number to a single item in one inventory organization. If the same item is also in a different inventory organization, you must assign that item a unique lot number. The following table provides an
example of how lot numbers are generated when uniqueness control is applied across items, both within and across inventory organizations.

<table>
<thead>
<tr>
<th>Within Inventory Organization</th>
<th>Across Inventory Organizations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item AS100 (printer) / Lot LN100</td>
<td>Item AS100 (printer) / Lot LN300</td>
</tr>
<tr>
<td>Item AS101 (laptop computer) / Lot LN200</td>
<td>Item AS101 (laptop computer) / Lot LN400</td>
</tr>
</tbody>
</table>

FAQs for Manage Inventory Organizations

**What happens if I select the Supplier item sourcing type for replenishment?**

Items are replenished from an external supplier.

**What happens if I create an inventory organization as a logistics services organization?**

The inventory organization is not costed, and shipment lines from different logistics service provider customers cannot be packed in the same packing unit.

Define Facilities: Manage Item Organizations

Item Organization: Explained

An item organization defines an item when inventory balances are not stored and inventory storage or inventory movement is not reflected in the Oracle Fusion Applications. For example, you would use an item organization in a retail scenario, if you need to know the items that are listed by and sold through each retail outlet even though inventory and transactions are recorded in another system. In Oracle Fusion Customer Relationship Management (CRM), item organizations are used to define sales catalogs.

**Note**

- Items belong to an item organization.

- Item attributes that are associated with financial and accounting information are hidden from the item if it exists within the item organization.

- Item organizations can be changed by administrators to an inventory organization by updating the necessary attributes. There is no difference in the way items are treated in these two types of organizations except that there cannot be any financial transactions in the downstream applications for items that are assigned to an item organization.

Item Master Organization: Explained

An item master organization lists and describes items that are shared across several inventory organizations or item organization.
The following example shows the choice between inventory organizations that track inventory transactions, stored in two warehouses, and item organizations that just track items, listed in two sales catalogs.

For the most efficient processing, you should:

- Have a single item master
- Include an item and its definition of form, fit, and function only once in the item master
- Separate the item master organization from organizations that store and transact items

Note

Oracle Fusion allows multiple item masters, however, use this capability cautiously. If you acquire a company, there may be value in allowing the old item master to exist for a transition period. If you manage your subsidiaries as separate businesses, there may be reduced value in a single item master.
Common Applications Configuration:
Define Security for Supply Chain Management

Security Tasks: Highlights

Security tasks include the following.

- Security setup
- Security implementation and administration

Note

Security setup and administration tasks typically use integrated user interface pages that are provided by the following products.

- Oracle Identity Manager (OIM)
- Oracle Authorization Policy Manager (APM)
- Oracle Fusion Human Capital Management (HCM) products
- Oracle Application Access Control Governor (AACG) in Governance, Risk, and Compliance Controls (GRCC)

Set Up the IT Security Manager Job Role

Provision the IT Security Manager job role with roles for user and role management.

- Using the OIM Administrator user name and password, sign in to Oracle Identity Manager (OIM). Refer to the Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Identity Management (Oracle Fusion Applications Edition).

See: Creating Users and Groups for Oracle Identity Manager
• Open the IT Security Manager job role's attributes and use the Hierarchy tab to add the User Identity Administrators role and the Role Administrators role in the OIM Roles category using the Add action. Use the Delegated Administration menu to search for the Xellerate Users organization and assign it to the IT Security Manager role. Refer to the Oracle Fusion Middleware User's Guide for Oracle Identity Manager.

See: User Management Tasks

Prerequisite Tasks for Security Administration

Sign into Oracle Fusion Applications for the first time with the Installation Super User account to synchronize LDAP users with HCM user management and create an IT security manager user account and provision it with the IT Security Manager role.

• The super user account is established during installation. Refer to the Oracle Fusion Applications Installation Guide.

See: Oracle Identity and Policy Management Configuration Parameters

• Synchronize LDAP users with HCM user management by performing the Run User and Roles Synchronization Process task. Monitor completion of the predefined Enterprise Scheduler job SyncRolesJob.

• Create a user account and provision it with roles. Refer to the Oracle Fusion Middleware User's Guide for Oracle Identity Manager.

See: Managing Users

• As a security guideline, provision a dedicated security professional with the IT Security Manager role as soon as possible after initial security setup and revoke that role from users provisioned with the Application Implementation Consultant role. If entitled to do so, see Security Tasks and Oracle Fusion Applications: How They Fit Together for details about provisioning the IT security manager.

Required Security Administration Tasks

Establish at least one implementation user and provision that user with sufficient access to set up the enterprise for all integrated Oracle Fusion Middleware and all application pillars or partitions.

• Perform the initial security tasks. If entitled to do so, see Initial Security Administration: Critical Choices.

• Sign in to Oracle Fusion Applications using the IT security manager's or administrator's user name and password, and create and provision users who manage your implementation projects and set up enterprise structures by performing the Create Implementation Users task. Refer to the Oracle Fusion Middleware User's Guide for Oracle Identity Manager.

See: User Management Tasks

• Create a data role for implementation users who will set up HCM that grants access to data in secured objects required for performing HCM setup steps. Provision the implementation user with this View All data role. See "Creating an HCM Data Role: Worked Example."
• For an overview of security tasks from the perspective of an applications administrator, refer to the Oracle Fusion Applications Administrator’s Guide

See: Securing Oracle Fusion Applications

Optional Security Administration Tasks

Once initial security administration is complete and your enterprise is set up with structures such as business units, additional security administration tasks are optional and based on modifying and expanding the predefined security reference implementation to fit your enterprise. See points to consider for defining security, data security and trading partner security after enterprise setup.

• Create users. Refer to the Oracle Fusion Middleware User’s Guide for Oracle Identity Manager.

See: Creating Users

• Provision users with roles. Refer to the Oracle Fusion Middleware User’s Guide for Oracle Identity Manager.

See: Adding and Removing Roles

• You manage users and job roles, including data and abstract roles, in Oracle Identity Management user interface pages. Refer to the Oracle Fusion Middleware User’s Guide for Oracle Identity Manager.

See: User Interfaces


See: Managing Oracle Fusion Applications Data Security Policies

• You manage role provisioning rules in Human Capital Management (HCM). If entitled to do so, see Role Mappings: Explained.

• For a complete description of the Oracle Fusion Applications security reference implementation, see the Oracle Fusion Applications Security Reference Manuals for each offering.

See: Oracle Fusion Applications Security Guide

• Since security in Oracle Fusion Applications is based on integrations with Oracle Identity Management in Fusion Middleware, security features in the database, and Governance, Risk, Compliance, and Controls, additional resources in support of performing security tasks include the following.

• Authorization Policy Manager (APM) is available in Oracle Fusion Applications through integration with Oracle Identity Management

See: Getting Started With Oracle Authorization Policy Manager

• Oracle Identity Management (OIM) is available in Oracle Fusion Applications through integration with Oracle Fusion Middleware. Identity management in Oracle Fusion Application involves creating and managing user identities, creating and linking user accounts, managing user access control through user role assignment, managing enterprise roles, and managing workflow approvals and delegated administration.

See: Oracle Fusion Middleware User’s Guide for Oracle Identity Manager

• Oracle Fusion Applications is certified to integrate with Applications Access Controls Governor (AACG) in the Oracle Governance, Risk and Compliance Controls (GRCC) suite to ensure effective segregation of duties (SOD).

See: Oracle Application Access Controls Governor Users Guide
See: Oracle Application Access Controls Governor Implementation Guide

• Configure and manage auditing. Refer to the Oracle Fusion Middleware Application Security Guide.

See: Configuring and Managing Auditing

Defining Security After Enterprise Setup: Points to Consider

After the implementation user has set up the enterprise, further security administration depends on the requirements of your enterprise.

The Define Security activity within the Information Technology (IT) Management business process includes the following tasks.

• Import Worker Users
• Import Partner Users
• Manage Job Roles
• Manage Duties
• Manage Application Access Controls

If no legacy users, user accounts, roles, and role memberships are available in the Lightweight Directory Access Protocol (LDAP) store, and no legacy workers are available in Human Resources (HR), the implementation user sets up new users and user accounts and provisions them with roles available in the Oracle Fusion Applications reference implementation.

If no legacy identities (workers, suppliers, customers) exist to represent people in your enterprise, implementation users can create new identities in Human
Before Importing Users

Oracle Identity Management (OIM) handles importing users.

If legacy employees, contingent workers, and their assignments exist, the HCM Application Administrator imports these definitions by performing the Load Batch Data task. If user and role provisioning rules have been defined, the Load Batch Data process automatically creates user and role provisioning requests as the workers are created.

Once the enterprise is set up, performing the Load Batch Data task populates the enterprise with HR workers in records linked by global user ID (GUID) to corresponding user accounts in the LDAP store. If no user accounts exist in the LDAP store, the Load Batch Data task results in new user accounts being created. Worker email addresses as an alternate input for the Load Batch Data task triggers a search of the LDAP for user GUIDs, which may perform more slowly than entering user names.

In the security reference implementation, the HCM Application Administrator job role hierarchy includes the HCM Batch Data Loading Duty role, which is entitled to import worker identities. This entitlement provides the access necessary to perform the Load Batch Data task in HCM.

Note

The Import Person and Organization task in the Define Trading Community Import activity imports the following resources, creates users, and links the resources to users for use in CRM.

- Internal employees
- Contingent workers
- External partner contacts
- Partner companies
- Legal entities
- Customers
- Consumers

If role provisioning rules have been defined, the Import Person and Organization task automatically provisions role requests as the users are created.

Import Users

If legacy users (identities) and user accounts exist outside the LDAP store that is being used by the Oracle Fusion Applications installation, the IT security manager has the option to import these definitions to the LDAP store by performing the Import Worker Users and Import Partner Users tasks.

If no legacy users or user accounts can be imported or exist in an LDAP repository accessible to Oracle Identity Management (OIM), the IT security manager creates users manually in OIM or uses the Load Batch Data task to create users from imported HR workers.
Once users exist, their access to Oracle Fusion Applications is dependent on the roles provisioned to them in OIM or Human Capital Management. Use the Manage HCM Role Provisioning Rules task to define rules that determine what roles are provisioned to users.

Importing user identities from other applications, including other Oracle Applications product lines, is either a data migration or manual task. Migrating data from other Oracle Applications includes user data. For more information about importing users, see the Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager.

In the security reference implementation, the IT Security Manager job role hierarchy includes the HCM Batch Data Loading Duty and the Partner Account Administration Duty. These duty roles provide entitlement to import or create users. The entitlement Load Batch Data provides the access necessary to perform the Import Worker Users task in OIM. The entitlement Import Partner entitlement provides the access necessary to perform the Import Partner Users task in OIM.

**Manage Job Roles**

Job and abstract roles are managed in OIM. This task includes creating and modifying job and abstract roles, but not managing role hierarchies of duties for the jobs.

**Note**

Manage Job Roles does not include provisioning job roles to users. Provisioning users is done in OIM, HCM, CRM or Oracle Fusion Supplier Portal.

Roles control access to application functions and data. Various types of roles identify the functions performed by users.

The Oracle Fusion Applications security reference implementation provides predefined job and abstract roles. In some cases, the jobs defined in your enterprise may differ from the predefined job roles in the security reference implementation. The predefined roles and role hierarchies in Oracle Fusion may require changes or your enterprise may require you to create new roles. For example, you need a job role for a petty cash administrator, in addition to an accounts payable manager. The security reference implementation includes a predefined Accounts Payable Manager, and you can create a petty cash administrator role to extend the reference implementation.

In the security reference implementation, the IT Security Manager job role hierarchy includes the Enterprise Role Management Duty role, which is entitled to manage job and abstract roles (the entitlement is Manage Enterprise Role). This entitlement provides the access necessary to perform the Manage Job Roles task in OIM.

**Manage Duties**

A person with a job role must be able to perform certain duties. In the Oracle Fusion Applications security reference implementation, enterprise roles inherit duties through a role hierarchy. Each duty corresponds to a duty role. Duty roles
specify the duties performed within applications and define the function and data access granted to the enterprise roles that inherit the duty roles.

Managing duties includes assigning duties to job and abstract roles in a role hierarchy using Authorization Policy Manager (APM). If your enterprise needs users to perform some actions in applications coexistent with Oracle Fusion applications, you may wish to remove the duty roles that enable those actions. For details about which duty roles are specific to the products in an offering, see the Oracle Fusion Applications Security Reference Manual for each offering.

OIM stores the role hierarchy and the spanning of roles across multiple pillars or logical partitions of applications.

In cases where your enterprise needs to provide access to custom functions, it may be necessary to create or modify the duty roles of the reference implementation.

Tip
As a security guideline, use only the predefined duty roles, unless you have added new applications functions. The predefined duty roles fully represent the functions and data that must be accessed by application users and contain all appropriate entitlement. The predefined duty roles are inherently without segregation of duty violations of the constraints used by the Application Access Controls Governor.

In the security reference implementation, the IT Security Manager job role hierarchy includes the Application Role Management Duty role, which is entitled to manage duty roles (the entitlement is Manage Application Role). This entitlement provides the access necessary to perform the Manage Duties task in APM.

Note
Product family administrators are not entitled to create role hierarchies or manage duty roles and must work with the IT security manager to make changes such as localizing a duty role to change a role hierarchy. Setup for localizations is documented in HCM documentation.

Manage Application Access Controls

Prevent or limit the business activities that a single person may initiate or validate by managing segregation of duties policies in the Application Access Controls Governor (AACG).

Note
In AACG, segregation of duties policies are called access controls or segregation of duties controls.

In the security reference implementation, the IT Security Manager job role hierarchy includes the Segregation of Duties Policy Management Duty role, which is entitled to manage segregation of duties policies (the entitlement is Manage Segregation of Duties Policy). This entitlement provides the access necessary to perform the Manage Application Access Controls task in AACG.
Security Tasks and Oracle Fusion Applications: How They Fit Together

The major security tasks and their order within the context of an overall Oracle Fusion Applications implementation extend from security setup through production deployment audits.

The Oracle Fusion business process model (BPM) provides a sequence of security implementation tasks that includes the following.

- Security setup (Define Common Applications Configuration activity)
  - Define Implementation Users task group (optional)
    - Create Implementation Users task
    - Create Data Role for Implementation Users task
    - Provision Roles to Implementation Users task
  - Define security - tasks vary depending on deployed Oracle Fusion product family
    - Revoke Data Role from Implementation Users task
    - Import Worker Users task
    - Import Partner Users task
    - Manage Duties task
    - Manage Job Roles task
    - Manage Application Access Controls task
    - Define Automated Governance, Risk, and Performance Controls activity
      - Manage Application Access Controls task (AACG settings)
      - Manage Application Preventive Controls task
      - Manage Application Transaction Controls task
      - Manage Application Configuration Controls task
  - User and role provisioning tasks
    - Implement Role Request and Provisioning Controls activity
      - Import Worker Users task
      - Import Partner Users task
      - Self Request User Roles task
      - Approve User and Role Provisioning Requests task
• Assign User Roles task
• Manage Supplier User Roles and User Role Usages task
• Map and Synchronize User Account Details task
• Tasks for viewing account details for self or others
• Tasks for applying and managing various role provisioning rules
• Tasks for running synchronization processes

• Security implementation and ongoing maintenance after setup (Manage IT Security activity)

• Implement Function Security Controls
  • Create Job Role task
  • Import Worker Users task
  • Import Partner Users task
  • Manage Duties task
  • Manage Job Roles task
  • Manage Users task

• Implement Data Security Controls
  • Manage Data Security Policies task
  • Manage Role Templates task
  • Manage Encryption Keys task
  • Manage Segment Security task
  • Manage Data Access Sets task
  • Define Security Profiles task group

• Auditing tasks
  • Manage Security Audit, Compliance and Reporting activity
  • Manage Application Access Controls task

---

**Note**
Go live deployment does not require lockdown or specific security tasks because security is enforced across the test to production information life cycle.

**Required Roles**
The Oracle Fusion Applications installation process sets up a single, super user provisioned with the following enterprise roles.
• Application Implementation Consultant
• IT Security Manager
• Application Administrators for the provisioned products

Initial security administration includes provisioning the IT Security Manager role with Oracle Identity Management (OIM) roles that carry entitlement for user and role management.

• Identity User Administrator
• Role Administrator

Additionally, the Xellerate Users organization must be assigned to the IT Security Manager role.

**Important**

As a security guideline, provision a dedicated security professional with the IT Security Manager role at the beginning of an implementation, and revoke that role from users provisioned with the Application Implementation Consultant role.

**Tools Used to Perform Security Tasks**

Security tasks are supported by tools within both Oracle Fusion Applications and Oracle Fusion Middleware.

The figure lists the tasks associated with each of the integrated products and pillars of an Oracle Fusion Applications deployment.
Security Tasks: Overview

Security tasks span multiple business processes and are performed by various roles using numerous integrated tools. The following table shows the business process model (BPM) tasks and tools used to support securing Oracle Fusion Applications.

<table>
<thead>
<tr>
<th>Example Task</th>
<th>Oracle BPM Task</th>
<th>Supporting Tools</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>View duty roles inherited by a job role</td>
<td>Manage Duties</td>
<td>• Authorization Policy Manager (APM)</td>
<td>Each logical partition or pillar contains a collection of application roles, and function and data security policies.</td>
</tr>
<tr>
<td>View entitlement or policies carried by a job role</td>
<td>Manage Duties</td>
<td>• APM</td>
<td>In LDAP, the policy store stores application roles and the identity store stores enterprise roles.</td>
</tr>
<tr>
<td>Add a job role to a role hierarchy</td>
<td>Manage Job Roles</td>
<td>• Oracle Identity Management (OIM)</td>
<td>The identity store in LDAP stores enterprise roles.</td>
</tr>
<tr>
<td>Add a duty role to a role hierarchy</td>
<td>Manage Duties</td>
<td>• APM</td>
<td>LDAP stores the role hierarchy and the spanning of roles across multiple pillars or logical partitions.</td>
</tr>
<tr>
<td>Create a hierarchy of enterprise (abstract, job, data) roles</td>
<td>Manage Job Roles</td>
<td>• OIM</td>
<td>The identity store in LDAP stores enterprise roles.</td>
</tr>
<tr>
<td>Create a hierarchy of (application) duty roles</td>
<td>Manage Duties</td>
<td>• APM</td>
<td>The policy store stores duty roles. The identity store stores enterprise roles. Some duty roles may enable actions and their associated users interface features that your enterprise does not want users to perform in Oracle Fusion applications.</td>
</tr>
<tr>
<td>Create a new job role</td>
<td>Manage Job Roles</td>
<td>• OIM</td>
<td>All functions and actions in Oracle Fusion Applications that need to be secured are covered by the reference implementation. In some cases, especially with function customizations, a new duty role may be needed.</td>
</tr>
<tr>
<td>Task</td>
<td>Manage or Use</td>
<td>Create or Access</td>
<td>Details</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>View Segregation of Duties (SOD) policies respected by a duty role</td>
<td>Manage Application Access Controls</td>
<td>• Application Access Controls Governor (AACG) in Governance, Risk, and Compliance Controls (GRCC)</td>
<td>The Security Reference Manuals (SRM) document the segregation of duties (SOD) policies respected within each job role</td>
</tr>
<tr>
<td>View SOD policy violations carried by the duty roles inherited by a job role</td>
<td>Manage Application Access Controls</td>
<td>• AACG in GRCC</td>
<td>The SRM documents the SOD conflicts for each job role</td>
</tr>
<tr>
<td>View SOD policy violations</td>
<td>Manage Segregation of Duties Policies</td>
<td>• AACG in GRC</td>
<td></td>
</tr>
</tbody>
</table>
| View the data security policies carried by a job, abstract, and data roles | Manage Data Security Policies                                                | • APM                                                                           | Oracle Fusion Data Security stores data security policies in the policy store.  
Data security can also be defined in application pages provided by Oracle Middleware Extensions for Applications (FND) |
| Create and update HCM security profiles                             | Manage Data Role and Security Profiles                                       | • Oracle Fusion HCM                                                             | This task does not include assigning data roles to the users, which is supported by user provisioning tasks. |
| Create (generate) a data role                                       | 1. Manage Role Templates  
2. Manage Data Roles and Security Profiles                                   | • APM                                                                           | Data roles are generated automatically based on data role templates and enterprise setup. Changes to data role templates generate new or changed data roles.  
Create data roles in HCM using the Manage Data Roles and Security Profiles task. |
<p>| Create a new data security policy (not through generated data roles based on data role templates or HCM security profiles) | Manage Data Security Policies                                                | • APM                                                                           | Data security can also be defined in application pages provided by Oracle Middleware Extensions for Applications (FND) |
| View data role templates defined by a product                       | Manage Role Templates                                                        | • APM                                                                           |                                                                      |
| Create or edit an existing data role template                       | Manage Role Templates                                                        | • APM                                                                           |                                                                      |</p>
<table>
<thead>
<tr>
<th>Secure common objects such as attachment categories or profile options</th>
<th>Manage Data Security Policies</th>
<th>• APM</th>
<th>Data security can also be defined in application pages provided by Oracle Middleware Extensions for Applications (FND)</th>
</tr>
</thead>
<tbody>
<tr>
<td>View, create, update encryption keys used to secure attributes of personally identifiable information</td>
<td>Manage Encryption Keys</td>
<td>• Oracle Fusion Payments</td>
<td></td>
</tr>
<tr>
<td>View, create, update Data Access Sets used to secure Ledgers and Ledger Sets</td>
<td>Manage Data Access Sets</td>
<td>• Oracle Fusion General Ledger</td>
<td></td>
</tr>
<tr>
<td>View, create, update accounting flexfield segment security rules</td>
<td>Manage Security Segments</td>
<td>• Oracle Fusion General Ledger</td>
<td></td>
</tr>
<tr>
<td>View or update the set of job roles that can be provisioned to supplier users</td>
<td>Manage Supplier User Role</td>
<td>• Supplier Portal • Sourcing</td>
<td>These tools are in the Oracle Fusion Procurement product family</td>
</tr>
<tr>
<td>Determine the supplier job roles that the supplier self service administrator can provision to supplier users</td>
<td>Manage Supplier User Role Usages</td>
<td>• Supplier Portal • Sourcing</td>
<td>These tools are in the Oracle Fusion Procurement product family</td>
</tr>
<tr>
<td>Set default supplier job roles based on the set of supplier roles that are defined by performing the Manage Supplier User Roles task</td>
<td>Manage Supplier User Role Usages</td>
<td>• Supplier Portal • Sourcing</td>
<td>These tools are in the Oracle Fusion Procurement product family</td>
</tr>
<tr>
<td>Create a new implementation user</td>
<td>Create Implementation Users</td>
<td>• OIM</td>
<td></td>
</tr>
<tr>
<td>Import legacy users</td>
<td>• Import Worker Users • Import Partner Users</td>
<td>• OIM</td>
<td></td>
</tr>
<tr>
<td>Create a new user</td>
<td>Manage Users</td>
<td>• HCM</td>
<td>HCM creates a new user and identity when a new worker is created. The Hire Employee and Add Contingent Worker tasks also result in new user creation requests. Creating a new user automatically triggers role provisioning requests based on role provisioning rules.</td>
</tr>
</tbody>
</table>
| Provision roles to a user | 1. Provision Roles to Implementation Users  
2. Manage Users | • OIM  
• Oracle Fusion HCM  
• Oracle Fusion CRM  
• Oracle Fusion Suppliers | Implementation users are provisioned through OIM since HCM is not setup at the start of the implementation. The Provision Roles to Implementation Users is not needed once implementation is complete.  
Once HCM is setup, HCM is used to provision roles to non-implementation users by performing the Manage Users task. Human Resources (HR) transaction flows such as Hire and Promote also provision roles.  
Once supplier users are setup, Supplier Model can be used by internal users to maintain supplier user accounts or supplier users can maintain their accounts in Supplier Portal. |
|---|---|---|---|
| View the job, abstract, and data roles provisioned to a user | 1. Manage Users  
2. Manage User Principal  
3. Provision Roles to Implementation Users | • Human Capital Management (HCM)  
• OIM | LDAP stores users, roles and provisioning information.  
The Manage User Principal and Provision Roles to Implementation Users tasks are not needed once implementation is complete. |
| Revoke role from user. | Manage Users | • HCM | You can revoke roles from various Human Resources task flows, the HCM Manage Users task and OIM. User termination includes role revocation. |
| Approve role provisioning or user account request. | Approve User and Role Provisioning Requests | • OIM |  |
| View audit logs | Not applicable | • Oracle Enterprise Manager | Viewing audit logs is a Oracle Fusion Middleware function and not represented by an Oracle Fusion Applications BPM task. |
For more information about provisioning identities and configuring audit policies, see the Oracle Fusion Applications Administrator’s Guide.

There may be more than one navigation path to the graphical user interface in which the task is performed. You can access most security tasks by starting in the Setup and Maintenance Overview page and searching for security tasks and task lists.

Define Data Security

Data Security: Explained

By default, users are denied access to all data.

Data security makes data available to users by the following means.

- Policies that define grants available through provisioned roles
- Policies defined in application code

You secure data by provisioning roles that provide the necessary access. Enterprise roles provide access to data through data security policies defined for the inherited application roles.

When setting up the enterprise with structures such as business units, data roles are automatically generated that inherit job roles based on data role templates. Data roles also can be generated based on HCM security profiles. Data role templates and HCM security profiles enable defining the instance sets specified in data security policies.

When you provision a job role to a user, the job role implicitly limits data access based on the data security policies of the inherited duty roles. When you provision a data role to a user, the data role explicitly limits the data access of the inherited job role to a dimension of data.

Data security consists of privileges conditionally granted to a role and used to control access to the data. A privilege is a single, real world action on a single business object. A data security policy is a grant of a set of privileges to a principal on an object or attribute group for a given condition. A grant authorizes a role, the grantee, to actions on a set of database resources. A database resource is an object, object instance, or object instance set. An entitlement is one or more allowable actions applied to a set of database resources.

Data is secured by the following means.

<table>
<thead>
<tr>
<th>Data security feature</th>
<th>Does what?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data security policy</td>
<td>Grants access to roles by means of entitlement</td>
</tr>
<tr>
<td>Role</td>
<td>Applies data security policies with conditions to users through role provisioning.</td>
</tr>
<tr>
<td>Data role template</td>
<td>Defines the data roles generated based on enterprise setup of data dimensions such as business unit.</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HCM security profile</td>
<td>Defines data security conditions on instances of object types such as person records, positions, and document types without requiring users to enter SQL code</td>
</tr>
<tr>
<td>Masking</td>
<td>Hides private data on non-production database instances</td>
</tr>
<tr>
<td>Encryption</td>
<td>Scrambles data to prevent users without decryption authorization from reading secured data</td>
</tr>
</tbody>
</table>

The sets of data that a user can access via roles are defined in Oracle Fusion Data Security. Oracle Fusion Data Security integrates with Oracle Platform Security Services (OPSS) to entitle users or roles (which are stored externally) with access to data. Users are granted access through the entitlement assigned to the roles or role hierarchy with which the user is provisioned. Conditions are WHERE clauses that specify access within a particular dimension, such as by business unit to which the user is authorized.

**Data Security Policies**

Data security policies articulate the security requirement "Who can do What on Which set of data,” where ‘Which set of data’ is an entire object or an object instance or object instance set and ‘What’ is the object entitlement.

For example, accounts payable managers can view AP disbursements for their business unit.

<table>
<thead>
<tr>
<th>Who</th>
<th>can do</th>
<th>what</th>
<th>on which set of data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounts payable managers</td>
<td>view</td>
<td>AP disbursements</td>
<td>for their business unit</td>
</tr>
</tbody>
</table>

A data security policy is a statement in a natural language, such as English, that typically defines the grant by which a role secures business objects. The grant records the following.

- Table or view
- Entitlement (actions expressed by privileges)
- Instance set (data identified by the condition)

For example, disbursement is a business object that an accounts payable manager can manage by payment function for any employee expenses in the payment process.

**Note**

Some data security policies are not defined as grants but directly in applications code. The security reference manuals for Oracle Fusion Applications offerings differentiate between data security policies that define a grant and data security policies defined in Oracle Fusion applications code.
A business object participating in a data security policy is the database resource of the policy.

Data security policies that use job or duty roles refer to data security entitlement.

For example, the data security policy for the Accounts Payable Manager job role refers to the view action on AP disbursements as the data security entitlement.

**Important**

The duty roles inherited by the job role can be moved and job roles reassembled without having to modify the data security.

As a security guideline, data security policies based on user session context should entitle a duty role. This keeps both function and data security policies at the duty role level, thus reducing errors.

For example, a Sales Party Management Duty can update Sales Party where the provisioned user is a member of the territory associated with the sales account. Or the Sales Party Management Duty can update Sales Party where the provisioned user is in the management chain of a resource who is on the sales account team with edit access. Or the Participant Interaction Management Duty can view an Interaction where the provisioned user is a participant of the Interaction.

For example, the Disbursement Process Management Duty role includes entitlement to build documents payable into payments. The Accounts Payable Manager job role inherits the Disbursement Process Management Duty role. Data security policies for the Disbursement Process Management Duty role authorize access to data associated with business objects such as AP disbursements within a business unit. As a result, the user provisioned with the Accounts Payable Manager job role is authorized to view AP disbursements within their business unit.

A data security policy identifies the entitlement (the actions that can be made on logical business objects or dashboards), the roles that can perform those actions, and the conditions that limit access. Conditions are readable WHERE clauses. The WHERE clause is defined in the data as an instance set and this is then referenced on a grant that also records the table name and required entitlement.

**Data Roles**

Data roles are implemented as job roles for a defined set of data.

A data role defines a dimension of data within which a job is performed. The data role inherits the job role that describes the job. For example, a data role entitles a user to perform a job in a business unit.

The data role inherits abstract or job roles and is granted data security privileges. Data roles carry the function security privileges inherited from job roles and also the data security privilege granted on database objects and table rows.

For example, an accounts payables specialist in the US Business Unit may be assigned the data role Accounts Payables Specialist - US Business Unit. This data role inherits the job role Accounts Payables Specialist and grants access to transactions in the US Business Unit.
A data role may be granted entitlement over a set people.

For example, a Benefits Administrator A-E is allowed to administer benefits for all people that have a surname that begins with A-E.

Data roles are created using data role templates. You create and maintain data roles in the Authorization Policy Manager (APM). Use the Manage Data Roles and Security Profiles task to create and maintain HCM data roles in Oracle Fusion HCM.

**HCM Security Profiles**

HCM security profiles are used to secure HCM data, such as people and departments. You use HCM security profiles to generate grants for an enterprise role. The resulting data role with its role hierarchy and grants operates in the same way as any other data role.

For example, an HCM security profile identifies all employees in the Finance division.

Oracle Fusion Payroll uses HCM security profiles to secure project organizations. Applications outside of HCM can use the HCM Data Roles UI pages to give their roles access to HR people.

**Masking and Encryption**

Oracle Fusion Applications uses masking to protect sensitive data from view by unauthorized users. Encryption APIs mask sensitive fields in applications user interfaces. Additionally, Oracle Data Masking is available for masking data in non-production instances and Oracle Transparent Data Encryption is available for protecting data in transit or in backups independent of managing encryption keys.

**Defining Data Security After Enterprise Setup: Points to Consider**

After the implementation user has set up the enterprise, further security administration depends on the requirements of your enterprise.

The Define Data Security activity within the Information Technology (IT) Management business process includes the following tasks.

- Manage Data Access Sets
- Manage Segment Security
- Manage Role Templates
- Manage Data Security Policies
- Manage Encryption Keys

These tasks address data security administration. For information on using the user interface pages for setting up and managing data security, see the Oracle

Note

The Manage Data Role and Security Profiles task, and all other HCM security profile setup tasks are documented in Human Capital Management (HCM) documentation.

Manage Data Access Sets

Data access sets define a set of access privileges to one or more ledgers or ledger sets.

The information on ledgers that are attached to data access sets are secured by function security. Users must have access to the segment values associated with the data access sets to access the corresponding GL account.

In the security reference implementation, the IT Security Manager job role hierarchy includes the Data Access Administration Duty role, which is entitled to manage data access sets (the entitlement is Define General Ledger Data Access Set). This entitlement provides the access necessary to perform the Manage Data Access Sets task in General Ledger.

Manage Segment Security

Balancing or management segment values can secure data within a ledger.

Segment values are stored in GL_ACCESS_SET_ASSIGNMENTS and secured by restrictions, such as Exclude, on parameters that control the set of values that a user can use during data entry.

In the security reference implementation, the IT Security Manager job role hierarchy includes the Application Key Flexfield Administration Duty role, which is entitled to manage application key flexfields (the entitlement is Manage Application Key Flexfield). This entitlement provides the access necessary to perform the Manage Segment Security task in General Ledger.

Manage Role Templates

Data role templates automatically create or update data roles based on dimensions such as business unit. As an enterprise expands, data role templates trigger replication of roles for added dimensions. For example, when creating a new business unit, a data role template generates a new Accounts Payables Manager data role based on the Financials Common Module Template for Business Unit Security data role template.

In the security reference implementation, the IT Security Manager job role hierarchy includes the Application Role Management Duty role, which is entitled to manage data role templates (the entitlement is Manage Role Template). This entitlement provides the access necessary to perform the Manage Role Templates task in APM.
Manage Data Security Policies

Data security grants provisioned to roles are data security policies. The security reference implementation provides a comprehensive set of predefined data security policies and predetermined data security policies based on data role templates.

Data security policies are available for review in Authorization Policy Manager (APM). Data security policies are implemented by grants stored in Oracle Fusion Data Security (FND_GRANTS).

Data security policies secure the database resources of an enterprise. Database resources are predefined applications data objects and should not be changed. However, for cases where custom database resources must be secured objects, the IT security manager is entitled to manage database resources and create new data security policies.

Warning

Review but do not modify HCM data security policies in APM except as a custom implementation. Use the HCM Manage Data Role And Security Profiles task to generate the necessary data security policies and data roles.

Manage Encryption Keys

Create or edit encryption keys held in Oracle Wallet to secure Personally Identifiable Information (PII) attributes This task is only available when Payments is implemented.

In the security reference implementation, the IT Security Manager job role hierarchy includes the Payments Data Security Administration Duty role, which is entitled to manage encryption keys that secure PII (the entitlement is Manage Wallet). This entitlement provides the access necessary to perform the Manage Encryptions Keys task in Payments.

Data Security in the Security Reference Implementation: Explained

The reference implementation contains a set of data security policies that can be inspected and confirmed to be suitable or a basis for further implementation using the Authorization Policy Manager (APM).

The security implementation of an enterprise is likely a subset of the reference implementation, with the enterprise specifics of duty roles, data security policies, and HCM security profiles provided by the enterprise.
The business objects registered as secure in the reference implementation are database tables and views.

Granting or revoking object entitlement to a particular user or group of users on an object instance or set of instances extends the base Oracle Fusion Applications security reference implementation without requiring customization of the applications that access the data.

**Data Security Policies in the Security Reference Implementation**

The data security policies in the reference implementation entitle the grantee (a role) to access instance sets of data based on SQL predicates in a WHERE clause.

---

**Tip**

When extending the reference implementation with additional data security policies, identify instance sets of data representing the business objects that need to be secured, rather than specific instances or all instances of the business objects.

---

Predefined data security policies are stored in the data security policy store, managed in the Authorization Policy Manager (APM), and described in the Oracle Fusion Applications Security Reference Manual for each offering. A data security policy for a duty role describes an entitlement granted to any job role that includes that duty role.

---

**Warning**

Review but do not modify HCM data security policies in APM except as a custom implementation. Use the HCM Manage Data Role And Security Profiles task to generate the necessary data security policies and data roles.

---

The reference implementation only enforces a portion of the data security policies in business intelligence that is considered most critical to risk management without negatively affecting performance. For performance reasons it is not practical to secure every level in every dimension. Your enterprise may have a different risk tolerance than assumed by the security reference implementation.

**HCM Security Profiles in the Security Reference Implementation**

The security reference implementation includes some predefined HCM security profiles for initial usability. For example, a predefined HCM security profile allows line managers to see the people that report to them.

The IT security manager uses HCM security profiles to define the sets of HCM data that can be accessed by the roles that are provisioned to users

**Data Roles**

The security reference implementation includes no predefined data roles to ensure a fully secured initial Oracle Fusion Applications environment.
The security reference implementation includes data role templates that you can use to generate a set of data roles with entitlement to perform predefined business functions within data dimensions such as business unit. Oracle Fusion Payables invoicing and expense management are examples of predefined business functions. Accounts Payable Manager - US is a data role you might generate from a predefined data role template for payables invoicing if you set up a business unit called US.

HCM provides a mechanism for generating HCM related data roles.

## Securing Data Access: Points to Consider

Oracle Fusion Applications supports securing data through role-based access control (RBAC) by the following methods.

<table>
<thead>
<tr>
<th>Method of securing data</th>
<th>Reason</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data roles apply explicit data security policies on job and abstract roles</td>
<td>Appropriate for job and abstract roles that should only access a subset of data, as defined by the data role template that generates the data role or by HCM security profiles.</td>
<td>Accounts Payable Manager - US data role to provide an accounts payable manager in the US business unit with access to invoices in the US business unit.</td>
</tr>
<tr>
<td>Data security policies</td>
<td>Define data access for application roles and provide inheriting job and abstract roles with implicit data security</td>
<td>Projects</td>
</tr>
</tbody>
</table>

If a user has access to the same function through different roles that access different data sets, then the user has access to a union of those data sets.

When a runtime session is created, Oracle Platform Security Services (OPSS) propagates only the necessary user to role mapping based on Oracle Fusion Data Security grants. A grant can specify entitlement to the following.

- Specific rows of data (data object) identified by primary key
- Groups of data (instance set) based on a predicate that names a particular parameter
- Data objects or instance sets based on runtime user session variables

Data is either identified by the primary key value of the row in the table where the data is stored. Or data is identified by a rule (SQL predicate) applied to the WHERE clause of a query against the table where the data is stored.

### Grants

Oracle Fusion Data Security can be used to restrict the following.

- Rows that are returned by a given query based on the intended business operation
• Actions that are available for a given row

Grants control which data a user can access.

Note

Attribute level security using grants requires a data security policy to secure the attribute and the entitlement check enforces that policy.

A grant logically joins a user or role and an entitlement with a static or parameterized object instance set. For example, \texttt{REGION='WEST'} is a static object instance set and \texttt{REGION=\&GRANT\_ALIAS\_PARAMETER1} is a parameterized object instance set. In the context of a specific object instance, grants specify the allowable actions on the set of accessible object instances. In the database, grants are stored in \texttt{FND\_GRANTS} and object instance sets are stored in \texttt{FND\_OBJECT\_INSTANCE\_SETS}. Object access can be tested using the privilege check application programming interface (API).

**Securing a Business Object**

A business object is a logical entity that is typically implemented as a table or view, and corresponds to a physical database resource. The data security policies of the security reference implementation secure predefined database resources. Use the Manage Data Security Policies task to define and register other database resources.

Data security policies identify sets of data on the registered business object and the actions that may be performed on the business object by a role. The grant can be made by data instance, instance set, or at a global level.

Note

Use parameterized object instance sets whenever feasible to reduce the number of predicates the database parses and the number of administrative intervention required as static object instances sets become obsolete. In HCM, security profiles generate the instance sets.

**Manage Data Security Policies**

**Database Resources and Data Security Policies: How They Work Together**

A data security policy applies a condition and allowable actions to a database resource for a role. When that role is provisioned to a user, the user has access to data defined by the policy. In the case of the predefined security reference implementation, this role is always a duty role. Data roles generated to inherit the job role based on data role templates limit access to database resources in a particular dimension, such as the US business unit.

The database resource defines and instance of a data object. The data object is a table, view, or flexfield.
The following figure shows the database resource definition as the means by which a data security policy secures a data object. The database resource names the data object. The data security policy grants to a role access to that database resource based on the policy’s action and condition.

**Database Resources**

A database resource specifies access to a table, view, or flexfield that is secured by a data security policy.

- Name providing a means of identifying the database resource
- Data object to which the database resource points

**Data Security Policies**

Data security policies consist of actions and conditions for accessing all, some, or a single row of a database resource.

- Condition identifying the instance set of values in the data object
- Action specifying the type of access allowed on the available values

---

**Note**

If the data security policy needs to be less restrictive than any available database resource for a data object, define a new data security policy.

---

**Actions**

Actions correspond to privileges that entitle kinds of access to objects, such as view, edit, or delete. The actions allowed by a data security policy include all or a subset of the actions that exist for the database resource.
Conditions

A condition is either a SQL predicate or an XML filter. A condition expresses the values in the data object by a search operator or a relationship in a tree hierarchy. A SQL predicate, unlike an XML filter, is entered in a text field in the data security user interface pages and supports more complex filtering than an XML filter, such as nesting of conditions or sub queries. An XML filter, unlike a SQL predicate, is assembled from choices in the UI pages as an AND statement.

Tip

An XML filter can be effective in downstream processes such as business intelligence metrics. A SQL predicate cannot be used in downstream metrics.

Manage Role Templates

Data Role Templates: Explained

You use data role templates to generate data roles. You generate such data roles, and create and maintain data role templates in the Authorization Policy Manager (APM).

Note

HCM data roles are generated using the Manage Data Roles and Security Profiles task, which uses HCM security profiles, not data role templates, to define the data security condition.

The following attributes define a data role template.

- Template name
- Template description
- Template group ID
- Base roles
- Data dimension
- Data role naming rule
- Data security policies

The data role template specifies which base roles to combine with which dimension values for a set of data security policies. The base roles are the parent job or abstract roles of the data roles.

Note

Abstract, job, and data roles are enterprise roles in Oracle Fusion Applications. Oracle Fusion Middleware products such as Oracle Identity Manager (OIM) and Authorization Policy Manager (APM) refer to enterprise roles as external
The dimension expresses stripes of data, such as territorial or geographic information you use to partition enterprise data. For example, business units are a type of dimension, and the values picked up for that dimension by the data role template as it creates data roles are the business units defined for your enterprise. The data role template constrains the generated data roles with grants of entitlement to access specific data resources with particular actions. The data role provides provisioned users with access to a dimensional subset of the data granted by a data security policy.

An example of a dimension is a business unit. An example of a dimension value is a specific business unit defined in your enterprise, such as US. An example of a data security policy is a grant to access a business object such as an invoice with a view entitlement.

When you generate data roles, the template applies the values of the dimension and participant data security policies to the group of base roles.

The template generates the data roles using a naming convention specified by the template’s naming rule. The generated data roles are stored in the Lightweight Directory Access Protocol (LDAP) store. Once a data role is generated, you provision it to users. A user provisioned with a data role is granted permission to access the data defined by the dimension and data security grant policies of the data role template.

For example, a data role template contains an Accounts Payable Specialist role and an Accounts Payable Manager role as its base roles, and region as its dimension, with the dimension values US and UK. The naming convention is [base-role-name]:[DIMENSION-CODE-NAME]. This data role template generates four data roles.

- Accounts Payable Specialist - US (business unit)
- Accounts Payable Specialist - UK (business unit)
- Accounts Payable Manager - US (business unit)
- Accounts Payable Manager - UK (business unit)

Making Changes To Data Role Templates

If you add a base role to an existing data role template, you can generate a new set of data roles. If the naming rule is unchanged, existing data roles are overwritten.

If you remove a base role from a data role template and regenerate data roles, a resulting invalid role list gives you the option to delete or disable the data roles that would be changed by that removal.

Making Changes to Dimension Values

If you add a dimension value to your enterprise that is used by a data role template, you must regenerate roles from that data role template to create a
data role for the new dimension. For example if you add a business unit to your enterprise, you must regenerate data roles from the data role templates that include business unit as a dimension.

If you add or remove a dimension value from your enterprise that is used to generate data roles, regenerating the set of data roles adds or removes the data roles for those dimension values. If your enterprise has scheduled regeneration as an Oracle Enterprise Scheduler Services process, the changes are made automatically.

For information on working with data role templates, see the Oracle Fusion Middleware Administrator’s Guide for Authorization Policy Manager (Oracle Fusion Applications Edition).

Define Users

Securing Identities and Users: Points To Consider

Identity covers all aspects of an entity’s existence within the contexts in which it is used. The identity of an enterprise user consists of HR attributes, roles, resources, and relationships.

HR attributes include identifying information about a user that is relatively static and well understood, such as first and last name, title, and job function.

Roles are part of a user’s identity and define the user’s purpose and responsibilities.

Within identity management, resources define what a user can and does do. In an enterprise, this typically translates into what resources a user has access to, what privileges they have on that resource, and what they have been doing on that resource. Resources can be application accounts or physical devices such as laptops or access cards. The enterprise owns the resources, secures them, and manages access to the resources by managing the user’s identity and access.

Relationships establish the portion of user identities that involve organizational transactions such as approvals.

An Oracle Fusion Applications user and corresponding identity are usually created in a single transaction, such as when a worker is created in Human Resources (HR). That transaction automatically triggers provisioning requests for the user based on role provisioning rules.

User accounts for some identities that are not employees, such as partner contacts, may be created in a later transaction using an identity that is already created in the identity store. Supplier contacts are created in the Supplier Model, not HR.

Stores

Various locations store identity and user data.
Identity data consists of the following.

- HR person records
- Oracle Fusion Trading Community Model party records

In Oracle Fusion Applications, identities and users correspond one to one, but not all identities correspond to a user, and not all users are provisioned with an identity. Some identities stored in HR and Trading Community Model may not be provisioned to user accounts and therefore are not synchronized with Oracle Identity Management (OIM). For example, a contact for a prospective customer is an identity in Trading Community Model but may not be provisioned with a user account in OIM. Some users stored in the Lightweight Directory Access Protocol (LDAP) store may not be provisioned with identities. For example, system user accounts used to run Web services to integrate third party services with Oracle Fusion Applications are not associated with a person record in HR or Trading Community Model. Some identifying credentials such as name, department, e-mail address, manager, and location are stored with user data in the LDAP store.

Importing Users

You can import users or user attributes in bulk from existing legacy identity and user stores.

Your tasks may include the following.

- Create users in bulk
- Update specific attributes for all users, such as postal code
- Link users to HR or Trading Community Model persons
- Monitor progress of the import process
- Correct errors & re-import
- Export users in bulk
- Import and export users using a standard plain text data interchange format like Lightweight Data Interchange Format (LDIF)

You can reserve a specific user name not currently in use for use in the future, or release a reserved username from the reservation list and make it available for use. Between a user registration request and approved registration, Oracle Fusion Applications holds the requested user name on the reservation list, and releases the name if an error occurs in the self-registration process or the request is rejected. Self-registration processes check the reservation list for user name availability and suggest alternative names.

Provisioning Events

New identities, such as new hires, trigger user and role provisioning events. In addition to user creation tasks, other tasks, such as Promote Worker or Transfer
Worker, result in role provisioning and recalculation based on role provisioning rules.

When an identity’s attributes change, you may need to provision the user with different roles. Role assignments may be based on job codes, and a promotion triggers role provisioning changes. Even if the change in the identities attributes requires no role assignment change, such as with a name change, OIM synchronizes the corresponding user information in the LDAP store.

Deactivating or terminating an identity triggers revocation of some roles to end all assignments, but may provision new roles needed for activities, such as a pay stub review. If the corresponding user for the identity was provisioned with a buyer role, terminating the identity causes the user’s buyer record in Procurement to be disabled, just as the record was created when the user was first provisioned with the buyer role.

**Notifications and Audits**

Oracle Fusion Applications provides mechanisms for notifying and auditing requests or changes affecting identities and users.

Oracle Fusion Applications notifies requestors, approvers, and beneficiaries when a user account or role is provisioned. For example, when an anonymous user registers as a business-to-customer (B2C) user, the B2C user must be notified of the registration activation steps, user account, password and so on once the approver (if applicable) has approved the request and the user is registered in the system.

User ID and GUID attributes are available in Oracle Fusion Applications session information for retrieving authenticated user and identity data.

End user auditing data is stored in database WHO columns and used for the following activities.

- Setting up sign-in audit
- Using the application monitor
- Notifying of unsuccessful sign ins
- Sign-in audit reports

You can conduct real time audits that instantiate a runtime session and impersonate the target user (with the proxy feature) to test what a user has access to under various conditions such as inside or outside firewall and authentication level.

For information on configuring audit policies and the audit store, see the Oracle Fusion Applications Administrator’s Guide.

**Delegated Administration**

You can designate local administrators as delegated administrators to manage a subset of users and roles.
Delegated administrators can be internal or external persons who are provisioned with a role that authorizes them to handle provisioning events for a subset of users and roles.

For example, internal delegated administrators could be designated to manage users and roles at the division or department level. External delegated administrators could be designated to manage users and roles in an external organization such as a primary supplier contact managing secondary users within that supplier organization.

You can also define delegated administration policies based on roles. You authorize users provisioned with specific roles named in the policy to request a subset of roles for themselves if needed, such as authorizing a subset of roles for a subset of people. For example, the policy permits a manager of an Accounts Payables department to approve a check run administrator role for one of their subordinates, but prohibits the delegated administrator from provisioning a budget approver role to the subordinate.

**Credentials**

You activate or change credentials on users by managing them in Oracle Identity Management (OIM)

Applications themselves must be credentialed to access one another.

Oracle Fusion Applications distinguishes between user identities and application identities (APPID). Predefined application identities serve to authorize jobs and transactions that require higher privileges than users.

For example, a payroll manager may submit a payroll run. The payroll application may need access to the employee’s taxpayer ID to print the payslip. However, the payroll manager is not authorized to view taxpayer IDs in the user interface as they are considered personally identifiable information (PII).

Calling applications use application identities (APPID) to enable the flow of transaction control as it moves across trust boundaries. For example, a user in the Distributed Order Orchestration product may release an order for shipping. The code that runs the Pick Notes is in a different policy store than the code that releases the product for shipment. When the pick note printing program is invoked it is the Oracle Fusion Distributed Order Orchestration Application Development Framework (ADF) that is invoking the program and not the end user.

**Manage HCM Role Provisioning Rules**

**Role Provisioning and Deprovisioning: Explained**

A user’s access to data and functions depends on the user’s roles: users have one or more roles that enable them to perform the tasks required by their jobs or
Positions. Roles must be provisioned to users; otherwise, users have no access to data or functions.

**Role Provisioning Methods**

Roles can be provisioned to users:

- Automatically
- Manually, using delegated administration:
  - Users such as line managers and human resource specialists can provision roles manually to other users.
  - Users can request roles for themselves.

For both automatic and manual role provisioning, you create a role mapping to identify when a user becomes eligible for a role.

Oracle Identity Management (OIM) can be configured to notify users when their roles change; notifications are not issued by default.

**Role Types**

Data roles, abstract roles, and job roles can be provisioned to users. Roles available for provisioning include predefined roles, HCM data roles, and roles created using OIM.

**Automatic Role Provisioning**

A role is provisioned to a user automatically when at least one of the user’s assignments satisfies the conditions specified in the relevant role-mapping definition. The provisioning occurs when the assignment is either created or updated. For example, when a person is promoted to a management position, the line manager role is provisioned automatically to the person if an appropriate role mapping exists. Any change to a person’s assignment causes the person’s automatically provisioned roles to be reviewed and updated as necessary.

**Role Deprovisioning**

Automatically provisioned roles are deprovisioned automatically as soon as a user no longer satisfies the role-mapping conditions. For example, a line manager role that is provisioned to a user automatically is deprovisioned automatically when the user ceases to be a line manager.

Automatically provisioned roles can be deprovisioned manually at any time. Manually provisioned roles are deprovisioned automatically only when all of the user’s work relationships are terminated; in all other circumstances, users retain manually provisioned roles until they are deprovisioned manually.

**Changes to Assignment Managers**

When a person’s line manager is changed, the roles of both new and previous line managers are updated as necessary. For example, if the person’s new line manager now satisfies the conditions in the role mapping for the line manager
role, and the role is one that is eligible for autoprovisioning, then that role is provisioned automatically to the new line manager. Similarly, if the previous line manager no longer satisfies the conditions for the line manager role, then that role is deprovisioned automatically.

**Roles at Termination**

When a work relationship is terminated, all automatically provisioned roles for which the user does not qualify in other work relationships are deprovisioned automatically. Manually provisioned roles are deprovisioned automatically only if the user has no other work relationships; otherwise, the user retains all manually provisioned roles until they are deprovisioned manually.

Automatic deprovisioning can occur either as soon as the termination is submitted or approved or on the day after the termination date. The user who is terminating the work relationship selects the appropriate deprovisioning date.

Role mappings can provision roles to users automatically at termination. For example, the locally defined roles Retiree and Beneficiary could be provisioned to users at termination based on assignment status and person type values.

If a termination is later reversed, roles that were deprovisioned automatically at termination are reinstated and post-termination roles are deprovisioned automatically.

**Date-Effective Changes to Assignments**

Automatic role provisioning and deprovisioning are based on current data. For a future-dated transaction, such as a future promotion, role changes are identified and role provisioning occurs on the day the changes take effect, not when the change is entered. The process Send Pending LDAP Requests identifies future-dated transactions and manages role provisioning and deprovisioning at the appropriate time. Note that such role-provisioning changes are effective as of the system date; therefore, a delay of up to 24 hours may occur before users in other time zones acquire the access for which they now qualify.

**Role Mappings: Explained**

User access to data and functions is determined by abstract, job, and data roles, which are provisioned to users either automatically or manually. To enable a role to be provisioned to users, you define a relationship, known as a mapping, between the role and a set of conditions, typically assignment attributes such as department, job, and system person type. In a role mapping, you can select any role stored in the Lightweight Directory Access Protocol (LDAP) directory, including Oracle Fusion Applications predefined roles, roles created in Oracle Identity Management (OIM), and HCM data roles.

The role mapping can support:

- Automatic provisioning of roles to users
- Manual provisioning of roles to users
- Role requests from users
- Immediate provisioning of roles
Automatic Provisioning of Roles to Users

A role is provisioned to a user automatically if:

- At least one of the user’s assignments satisfies all conditions associated with the role in the role mapping.
- You select the Autoprovision option for the role in the role mapping.

For example, for the HCM data role Sales Manager Finance Department, you could select the Autoprovision option and specify the following conditions.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Finance Department</td>
</tr>
<tr>
<td>Job</td>
<td>Sales Manager</td>
</tr>
<tr>
<td>Assignment Status</td>
<td>Active</td>
</tr>
</tbody>
</table>

The HCM data role Sales Manager Finance Department is provisioned automatically to users with at least one assignment that satisfies all of these conditions.

Automatic role provisioning occurs as soon as the user is confirmed to satisfy the role-mapping conditions, which can be when the user’s assignment is either created or updated. The provisioning process also removes automatically provisioned roles from users who no longer satisfy the role-mapping conditions.

Note

The automatic provisioning of roles to users is effectively a request to OIM to provision the role. OIM may reject the request if it violates segregation-of-duties rules or fails a custom OIM approval process.

Manual Provisioning of Roles to Users

Users such as human resource (HR) specialists and line managers can provision roles manually to other users; you create a role mapping to identify roles that can be provisioned in this way.

Users can provision a role to other users if:

- At least one of the assignments of the user who is provisioning the role (for example, the line manager) satisfies all conditions associated with the role mapping.
- You select the Requestable option for the role in the role mapping.

For example, for the HCM data role Quality Assurance Team Leader, you could select the Requestable option and specify the following conditions.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager with Reports</td>
<td>Yes</td>
</tr>
<tr>
<td>Assignment Status</td>
<td>Active</td>
</tr>
</tbody>
</table>
Any user with at least one assignment that satisfies both of these conditions can provision the role Quality Assurance Team Leader manually to other users, who are typically direct and indirect reports.

If the user's assignment subsequently changes, there is no automatic effect on roles provisioned by this user to others; they retain manually provisioned roles until either all of their work relationships are terminated or the roles are manually deprovisioned.

Role Requests from Users

Users can request roles when reviewing their own account information; you create a role mapping to identify roles that users can request for themselves.

Users can request a role if:

- At least one of their own assignments satisfies all conditions associated with the role mapping.

- You select the **Self-requestable** option for the role in the role mapping.

For example, for the Expenses Reporting role you could select the **Self-requestable** option and specify the following conditions.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>ABC Department</td>
</tr>
<tr>
<td>System Person Type</td>
<td>Employee</td>
</tr>
<tr>
<td>Assignment Status</td>
<td>Active</td>
</tr>
</tbody>
</table>

Any user with at least one assignment that satisfies all of these conditions can request the role. The user acquires the role either immediately or, if approval is required, once the request is approved. Self-requested roles are classified as manually provisioned.

If the user's assignment subsequently changes, there is no automatic effect on self-requested roles. Users retain manually provisioned roles until either all of their work relationships are terminated or the roles are manually deprovisioned.

Immediate Provisioning of Roles

When you create a role mapping, you can apply autoprovisioning from the role mapping itself.

In this case, all assignments and role mappings in the enterprise are reviewed. Roles are:

- Provisioned immediately to all users who do not currently have roles for which they are eligible

- Deprovisioned immediately from users who are no longer eligible for roles that they currently have

Immediate autoprovisioning from the role mapping enables bulk automatic provisioning of roles to a group of users who are identified by the role-mapping conditions. For example, if you create a new department after a merger, you
can provision relevant roles to all users in the new department by applying autoprovisioning immediately.

To provision roles immediately to a single user, the user’s line manager or an HR specialist can autoprovision roles from that user’s account.

**Role-Mapping Names**

The names of role mappings must be unique in the enterprise. You are recommended to devise a naming scheme that reveals the scope of each role mapping. For example:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autoprovisioned Roles Sales Department</td>
<td>Mapping includes all roles provisioned automatically to anyone in the sales department</td>
</tr>
<tr>
<td>Benefits Specialist Autoprovisioned</td>
<td>Mapping defines the conditions for autoprovisioning the Benefits Specialist role</td>
</tr>
<tr>
<td>Line Manager Requestable Roles</td>
<td>Mapping includes all roles that a line manager can provision manually to direct and indirect reports</td>
</tr>
</tbody>
</table>

**Role Mappings: Examples**

Roles must be provisioned to users explicitly, either automatically or manually; no role is provisioned to a user by default. This topic provides some examples of typical role mappings to support automatic and manual role provisioning.

**Creating a Role Mapping for Employees**

You want all employees in your enterprise to have the Employee role automatically when they are hired. In addition, employees must be able to request the Expenses Reporting role when they need to claim expenses. Few employees will need this role, so you decide not to provision it automatically to all employees.

You create a role mapping called All Employees and enter the following conditions.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Person Type</td>
<td>Employee</td>
</tr>
<tr>
<td>Assignment Status</td>
<td>Active</td>
</tr>
</tbody>
</table>

In the role mapping you include the:

- Employee role, and select the **Autoprovision** option
- Expenses Reporting role, and select the **Self-requestable** option

You could create a similar role mapping for contingent workers called All Contingent Workers, where you would set the system person type to contingent worker.

**Note**
If the Employee and Contingent Worker roles are provisioned automatically, pending workers acquire them when their periods of employment or placements start. If they need roles before then, you create a separate role mapping for the pending worker system person type.

Creating a Role Mapping for Line Managers

Any type of worker can be a line manager in the sales business unit. You create a role mapping called Line Manager Sales BU and enter the following conditions.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Unit</td>
<td>Sales</td>
</tr>
<tr>
<td>Assignment Status</td>
<td>Active</td>
</tr>
<tr>
<td>Manager with Reports</td>
<td>Yes</td>
</tr>
</tbody>
</table>

You include the Line Manager role and select the Autoprovision option. This role mapping ensures that the Line Manager role is provisioned automatically to any worker with at least one assignment that matches the role-mapping conditions.

In the same role mapping, you could include roles that line managers in this business unit can provision manually to other users by selecting the roles and marking them as requestable. Similarly, if line managers can request roles for themselves, you could include those in the same role mapping and mark them as self-requestable.

Creating a Role Mapping for Retirees

Retirees in your enterprise need a limited amount of system access to manage their retirement accounts. You create a role mapping called All Retirees and enter the following conditions.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Person Type</td>
<td>Retiree</td>
</tr>
<tr>
<td>Assignment Status</td>
<td>Inactive</td>
</tr>
</tbody>
</table>

You include the locally defined role Retiree in the role mapping and select the Autoprovision option. When at least one of a worker’s assignments satisfies the role-mapping conditions, the Retiree role is provisioned to that worker automatically.

Creating a Role Mapping for Sales Managers

Grade 6 sales managers in the sales department need the Sales Manager role. In addition, sales managers need to be able to provision the Sales Associate role to other workers. You create a role mapping called Sales Managers Sales Department and enter the following conditions.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Sales</td>
</tr>
</tbody>
</table>
In the role mapping, you include the:

- Sales Manager role, and select the Autoprovission option
- Sales Associate role, and select the Requestable option

**Import Worker Users**

**Defining Security After Enterprise Setup: Points to Consider**

After the implementation user has set up the enterprise, further security administration depends on the requirements of your enterprise.

The Define Security activity within the Information Technology (IT) Management business process includes the following tasks.

- Import Worker Users
- Import Partner Users
- Manage Job Roles
- Manage Duties
- Manage Application Access Controls

If no legacy users, user accounts, roles, and role memberships are available in the Lightweight Directory Access Protocol (LDAP) store, and no legacy workers are available in Human Resources (HR), the implementation user sets up new users and user accounts and provisions them with roles available in the Oracle Fusion Applications reference implementation.

If no legacy identities (workers, suppliers, customers) exist to represent people in your enterprise, implementation users can create new identities in Human Capital Management (HCM), Supplier Portal, and Customer Relationship Management (CRM) Self Service, respectively, and associate them with users.

**Before Importing Users**

Oracle Identity Management (OIM) handles importing users.

If legacy employees, contingent workers, and their assignments exist, the HCM Application Administrator imports these definitions by performing the Load Batch Data task. If user and role provisioning rules have been defined, the Load Batch Data process automatically creates user and role provisioning requests as the workers are created.

Once the enterprise is set up, performing the Load Batch Data task populates the enterprise with HR workers in records linked by global user ID (GUID)
to corresponding user accounts in the LDAP store. If no user accounts exist in
the LDAP store, the Load Batch Data task results in new user accounts being
created. Worker email addresses as an alternate input for the Load Batch Data
task triggers a search of the LDAP for user GUIDs, which may perform more
slowly than entering user names.

In the security reference implementation, the HCM Application Administrator
job role hierarchy includes the HCM Batch Data Loading Duty role, which
is entitled to import worker identities. This entitlement provides the access
necessary to perform the Load Batch Data task in HCM.

---

**Note**

The Import Person and Organization task in the Define Trading Community
Import activity imports the following resources, creates users, and links the
resources to users for use in CRM.

- Internal employees
- Contingent workers
- External partner contacts
- Partner companies
- Legal entities
- Customers
- Consumers

If role provisioning rules have been defined, the Import Person and Organization
task automatically provisions role requests as the users are created.

---

**Import Users**

If legacy users (identities) and user accounts exist outside the LDAP store that
is being used by the Oracle Fusion Applications installation, the IT security
manager has the option to import these definitions to the LDAP store by
performing the Import Worker Users and Import Partner Users tasks.

If no legacy users or user accounts can be imported or exist in an LDAP
repository accessible to Oracle Identity Management (OIM), the IT security
manager creates users manually in OIM or uses the Load Batch Data task to
create users from imported HR workers.

Once users exist, their access to Oracle Fusion Applications is dependent on
the roles provisioned to them in OIM or Human Capital Management. Use the
Manage HCM Role Provisioning Rules task to define rules that determine what
roles are provisioned to users.

Importing user identities from other applications, including other Oracle
Applications product lines, is either a data migration or manual task. Migrating
data from other Oracle Applications includes user data. For more information
about importing users, see the Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager.

In the security reference implementation, the IT Security Manager job role hierarchy includes the HCM Batch Data Loading Duty and the Partner Account Administration Duty. These duty roles provide entitlement to import or create users. The entitlement Load Batch Data provides the access necessary to perform the Import Worker Users task in OIM. The entitlement Import Partner entitlement provides the access necessary to perform the Import Partner Users task in OIM.

**Manage Job Roles**

Job and abstract roles are managed in OIM. This task includes creating and modifying job and abstract roles, but not managing role hierarchies of duties for the jobs.

---

**Note**

Manage Job Roles does not include provisioning job roles to users. Provisioning users is done in OIM, HCM, CRM or Oracle Fusion Supplier Portal.

---

Roles control access to application functions and data. Various types of roles identify the functions performed by users.

The Oracle Fusion Applications security reference implementation provides predefined job and abstract roles. In some cases, the jobs defined in your enterprise may differ from the predefined job roles in the security reference implementation. The predefined roles and role hierarchies in Oracle Fusion may require changes or your enterprise may require you to create new roles. For example, you need a job role for a petty cash administrator, in addition to an accounts payable manager. The security reference implementation includes a predefined Accounts Payable Manager, and you can create a petty cash administrator role to extend the reference implementation.

In the security reference implementation, the IT Security Manager job role hierarchy includes the Enterprise Role Management Duty role, which is entitled to manage job and abstract roles (the entitlement is Manage Enterprise Role). This entitlement provides the access necessary to perform the Manage Job Roles task in OIM.

**Manage Duties**

A person with a job role must be able to perform certain duties. In the Oracle Fusion Applications security reference implementation, enterprise roles inherit duties through a role hierarchy. Each duty corresponds to a duty role. Duty roles specify the duties performed within applications and define the function and data access granted to the enterprise roles that inherit the duty roles.

Managing duties includes assigning duties to job and abstract roles in a role hierarchy using Authorization Policy Manager (APM). If your enterprise needs users to perform some actions in applications coexistent with Oracle Fusion
applications, you may wish to remove the duty roles that enable those actions. For details about which duty roles are specific to the products in an offering, see the Oracle Fusion Applications Security Reference Manual for each offering.

OIM stores the role hierarchy and the spanning of roles across multiple pillars or logical partitions of applications.

In cases where your enterprise needs to provide access to custom functions, it may be necessary to create or modify the duty roles of the reference implementation.

---

**Tip**

As a security guideline, use only the predefined duty roles, unless you have added new applications functions. The predefined duty roles fully represent the functions and data that must be accessed by application users and contain all appropriate entitlement. The predefined duty roles are inherently without segregation of duty violations of the constraints used by the Application Access Controls Governor.

---

In the security reference implementation, the IT Security Manager job role hierarchy includes the Application Role Management Duty role, which is entitled to manage duty roles (the entitlement is Manage Application Role). This entitlement provides the access necessary to perform the Manage Duties task in APM.

---

**Note**

Product family administrators are not entitled to create role hierarchies or manage duty roles and must work with the IT security manager to make changes such as localizing a duty role to change a role hierarchy. Setup for localizations is documented in HCM documentation.

---

**Manage Application Access Controls**

Prevent or limit the business activities that a single person may initiate or validate by managing segregation of duties policies in the Application Access Controls Governor (AACG).

---

**Note**

In AACG, segregation of duties policies are called access controls or segregation of duties controls.

---

In the security reference implementation, the IT Security Manager job role hierarchy includes the Segregation of Duties Policy Management Duty role, which is entitled to manage segregation of duties policies (the entitlement is Manage Segregation of Duties Policy). This entitlement provides the access necessary to perform the Manage Application Access Controls task in AACG.
Manage Users

Creating Users: Worked Example

You can create users by entering basic person and employment data. A user account is created automatically for a person when you create the user record. You can assign the users Oracle Fusion Human Capital Management (HCM) and non-HCM data roles, each providing access to specific functions and data. This example demonstrates how to create a user and assign roles to the user.

Note

This user management functionality is available for HCM Foundation and Oracle Fusion Workforce Directory Management (WDM) users only.

<table>
<thead>
<tr>
<th>Decisions to Consider</th>
<th>In this Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>For whom are you creating the user record?</td>
<td>Gail Williams</td>
</tr>
<tr>
<td>What is the user account name?</td>
<td>Same as the e-mail ID, <a href="mailto:gail.williams@vision.com">gail.williams@vision.com</a></td>
</tr>
<tr>
<td>Where is Gail employed?</td>
<td>Gail is an employee of Vision Corporation, and works in the Human Resources (HR) department in the Canada office.</td>
</tr>
<tr>
<td>What roles must be provisioned to Gail?</td>
<td>Autoprovision the employee role. Gail is responsible for processing workers’ expense claims so provision the role Expense Claims Administrator manually to Gail.</td>
</tr>
</tbody>
</table>

Prerequisites

1. Create a role mapping called All Employees and enter the following conditions.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Person Type</td>
<td>Employee</td>
</tr>
<tr>
<td>Assignment Status</td>
<td>Active</td>
</tr>
</tbody>
</table>

In the role mapping you include the:

- Employee role, and select the Autoprovision option
- Expense Claims Administrator role, and select the Self-requestable option

Creating a User

1. On the Search Person page, click the Create icon to open the Create User page.

2. Complete the fields, as shown in this table:
3. In the User Details region, leave the User Name field blank. The user name defaults to the user's e-mail ID.

4. In the Employment Information region, select the person type **Employee** and complete the fields as shown in the table:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Name</td>
<td>Williams</td>
</tr>
<tr>
<td>First Name</td>
<td>Gail</td>
</tr>
<tr>
<td>E-Mail</td>
<td><a href="mailto:gail.williams@vision.com">gail.williams@vision.com</a></td>
</tr>
<tr>
<td>Hire Date</td>
<td>4/12/11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legal Employer</td>
<td>Vision Corporation</td>
</tr>
<tr>
<td>Business Unit</td>
<td>Vision Canada</td>
</tr>
<tr>
<td>Department</td>
<td>Human Resources</td>
</tr>
</tbody>
</table>

**Assigning Roles to the User**

1. Click **Autoprovision Roles** to provision the employee role to the user.
2. Click **Add Role**.
3. Search for and select the **Expense Claims Administrator** role.
4. Click **Save and Close**. The user account is created and the roles are assigned to the user immediately.
Segregation of Duties: Explained

Segregation of duties (SOD) separates activities such as approving, recording, processing, and reconciling results so an enterprise can more easily prevent or detect unintentional errors and willful fraud. SOD policies, called access control policies in Application Access Controls Governor (AACG), exert both preventive and detective effects.

SOD policies constrain duties across roles so that unethical, illegal, or damaging activities are less likely. SOD policies express constraints among roles. Duty role definitions respect segregation of duties policies.

Application Access Controls Governor

You manage, remediate, and enforce access controls to ensure effective SOD using the Application Access Controls Governor (AACG) product in the Oracle Governance, Risk and Compliance Controls (GRCC) suite.

AACG applies the SOD policies of the Oracle Fusion Applications security reference implementation using the AACG Oracle Fusion Adapter.

AACG is integrated with Oracle Identity Management (OIM) in Oracle Fusion Applications to prevent SOD control violations before they occur by ensuring SOD compliant user access provisioning. SOD constraints respect provisioning workflows. For example, when provisioning a Payables role to a user, the SOD policy that ensures no user is entitled to create both an invoice and a payment prevents the conflicting roles from being provisioned. AACG validates the request to provision a user with roles against SOD policies and provides a remediating response such as approval or rejections if a violation is raised.

Use AACG to for the following.

- Define SOD controls at any level of access such as in the definition of an entitlement or role.
- Simulate what-if SOD scenarios to understand the effect of proposed SOD control changes.
- Use the library of built-in SOD controls provided as a security guideline.

**Managing Segregation of Duties**

SOD policies express incompatible entitlement or incompatible access points into an application. In GRCC, an access point is the lowest level access for a particular application. In GRCC, entitlement is a grouping of access points. As a security guideline, group the lowest level access points or define the SOD policy at the access level causing the least amount of change. Business activities are enabled at access points. In Oracle Fusion Applications, the hierarchy of access points in descending levels is users, roles, and entitlement.

---

**Note**

AACG entitlements are logical groupings of security objects that represent Oracle Fusion Application access points such as roles or entitlement.

---

**Note**

In AACG, segregation of duties policies are called access controls.

Oracle Fusion Applications does not predefine business logic for dealing with SOD conflicts. Oracle Fusion Applications does define a set of states where role requests are suspended pending resolution of SOD violations the role request introduces. In most cases, Oracle Fusion Applications invokes OIM to handle role requests. Enterprises define SOD resolution rules when defining SOD policy.

**Remediating Segregation of Duties Policy Violations**

The risk tolerance of your enterprise determines what duties must be segregated and how to address violations.

AACG assists in remediation of violations with a guided simulation that identifies corrective action. You determine the exact effects of role and entitlement changes prior to putting them into production, and adjust controls as needed.

For information on managing segregation of duties, see the Oracle Application Access Controls Governor Implementation Guide and Oracle Application Access Controls Governor User's Guide.

---

**Segregation of Duties in the Security Reference Implementation: Explained**

Segregation of duties (SOD) is a special case of function security enforcement. A segregation of duties conflict occurs when a single user is provisioned with a role or role hierarchy that authorizes transactions or operations resulting in the possibility of intentional or inadvertent fraud.
The predefined SOD policies result in duty separation with no inherent violations. For example, an SOD policy prevents a user from entitlement to create both payables invoices and payables payments.

However, the most common duties associated with some job and abstract roles could conflict with the predefined segregation of duties. A predefined role hierarchy or job or abstract role may include such common duties that are incompatible according to a segregation of duties policy. For example, the predefined Accounts Payable Supervisor job role includes the incompatible duties: Payables Invoice Creation Duty and Payables Payment Creation Duty.

Every single predefined duty role is free from an inherent segregation of duties violation. For example, no duty role violates the SOD policy that prevents a user from entitlement to both create payables invoices and payables payments.

Jobs in the reference implementation may contain violations against the implemented policies and require intervention depending on your risk tolerance, even if you define no additional jobs or SOD policies.

Provisioning enforces segregation of duties policies. For example, provisioning a role to a user that inherits a duty role with entitlement to create payables invoices enforces the segregation of duties policy applied to that duty role and ensures the user is not also entitled to create a payables payment. When a role inherits several duty rules that together introduce a conflict, the role is provisioned with a violation being raised in the Application Access Controls Governor (AACG). If two roles are provisioned to a user and introduce a segregation of duties violation, the violation is raised in AACG.

**Note**

SOD policies are not enforced at the time of role definition.

Aspects of segregation of duties policies in the security reference implementation involve the following:

- Application Access Controls Governor (AACG)
- Conflicts defined in segregation of duties policies
- Violations of the conflicts defined in segregation of duties policies

**Application Access Controls Governor (AACG)**

AACG is a component of the Governance, Risk, and Compliance Controls (GRCC) suite of products where segregation of duties policies are defined.

- Define SOD controls at any level of access such as in the definition of an entitlement or role.
- Simulate what-if SOD scenarios to understand the effect of proposed SOD control changes.
- Use the library of built-in SOD controls provided as a security guideline.

Your risk tolerance determines how many duties to segregate. The greater the segregation, the greater the cost to the enterprise in complexity at implementation and during maintenance. Balance the cost of segregation with the reduction of risk based on your business needs.
Conflicts

An intra-role conflict occurs when a segregation of duties policy expresses
constraints within the construct of a single role (entitlement and duties) that
creates violations.

Tip

As a security guideline, use only the predefined duty roles, unless you have
added new applications functions. The predefined duty roles fully represent
the functions and data that must be accessed by application users and contain
all appropriate entitlement. The predefined duty roles are inherently without
segregation of duty violations of the constraints used by the Application Access
Controls Governor.

Violations

A segregation of duties violation occurs when a policy is defined that allows a
segregation of duties conflict to occur.

Notifications report conflicts to the requester of the transaction that raised the
violation. Oracle Identity Management (OIM) shows the status of role requests
indicating if a segregation of duties violation has occurred.

For information on configuring audit policies, see the Oracle Fusion Applications
Administrator’s Guide.

For more information on managing segregation of duties, see the Oracle
Application Access Controls Governor Implementation Guide and Oracle
Application Access Controls Governor User’s Guide.

Defining Segregation of Duties Policies: Points To Consider

Segregation of duties (SOD) policies express incompatibilities enforced to control
access in defined contexts.

In Oracle Fusion Applications, SOD policies protect against the following
incompatibilities.

- Privilege X is incompatible with privilege Y
- Role A is incompatible with role B
- Any privileges in role A are incompatible with any privileges in role B.
- Privilege X is incompatible with any privileges in role B.

The following examples of SOD policies illustrate incompatible entitlement.

- No user should have access to Bank Account Management and Supplier
  Payments duties.
• No user should have access to Update Supplier Bank Account and Approve Supplier Invoice entitlement.

Data Contexts

You can extend SOD policies to control access to specific data contexts.

For example, no single individual must be able to source a supplier in a business unit and approve a supplier invoice in the same business unit.

Exclusion and Inclusion Conditions

SOD policies may include exclusion conditions to narrow the SOD scope and reduce false positive violations, or inclusion conditions to broaden the scope.

Conditions apply to access points globally, to policies, or to access paths defined by policies. Access path conditions can exclude a user from a role, an Oracle Fusion Applications entitlement from a role, or a permission from an Oracle Fusion Applications entitlement.

The following global exclusion conditions are predefined in Oracle Fusion Applications and available when creating SOD policies.

- User Status
- User Name
- Enterprise Role
- Action
- Business Unit
- Within Same Business Unit

Enforcement

Oracle Fusion Applications enforces SOD policies under the following circumstances.

- When granting entitlement to a role
- When provisioning a role to a user

For information on managing segregation of duties, see Oracle Application Access Controls Governor Implementation Guide and Oracle Application Access Controls Governor User's Guide.

Note

SOD policies are not enforced at the time of role definition.

A single SOD policy can include entitlement from multiple instances of a single enterprise resource planning environment. For example, one SOD policy is
Managing Segregation of Duties Risks and Violations: Critical Choices

You assess and balance the cost of duty segregation against reduction of risk based on the requirements of your enterprise.

The types of people who resolve SOD conflicts include the following.

- Administrator of an external program such as the Procurement Administrator for the supplier portal or the Partner Manager for the PRM Program
- Senior executive spanning multiple organizations in an enterprise with opposing interests
- Risk management professional implementing an Oracle Governance, Risk and Compliance Controls (GRCC) initiative
- Predefines a set of conditions and informs access provisioning staff to approve requests and prove the exception based on certain conditions
- Allows defining rules to route SOD violations for approval

You view and respond to risks and violations in the Application Access Controls Governor (AACG).

You may wish to override an SOD violation. For example, the Accounts Payable Supervisor includes incompatible duties to create both invoices and payments. When you provision this job role to a user, you may waive the violation in the AACG. You may waive the violation for the currently provisioned user, for the SOD policy that raised the violation, or for the SOD policy within a particular data set, such as a business unit.

The risk tolerance of your enterprise guides how you respond to conflicts. For example, a user may be provisioned with both the role of Order Manager and Shipping Agent. The Order Manger role entitles the user to enter orders, which could result in exploitation when filling shipping quotas. You can remove the entitlement to enter orders that the Order Manger job role inherits from the Orchestration Order Scheduling Duty role. Or you could segregate the shipping and order entry duties by defining an SOD policy that allows a user to have either job role but not both.

False Positives

False positives can be SOD policy violations that are not actually violations, or are violations within your risk tolerance and therefore do not require corrective action.

You can reduce false positives by the following methods.
• Define exclusion conditions that can be applied to individual or groups of policies.
• Define logically complex SOD policies that enforce more exacting specifications.
• Determine whether conflicts should be prevented, monitored, or subjected to approval during provisioning.

Path Level Detection

Conflict analysis detects a user’s multiple paths to one or more conflicting access points.

For example, a user may be able to reach a single access point through one or more roles, or by one entitlement leading to another through submenus to a function that represents a risk. The resulting conflict path shows if the conflict is generated by inappropriate role provisioning or configuration of applications. The audit shows the paths from any number of users to any number of access points involved in conflicts, which lets you visualize the root cause and remediate effectively.

AACG assigns one or more users to review all paths involved in a given conflict so that the entire conflict can be addressed in a coherent way.

Waiving or Accepting Violations

AACG lets you accept or waive a violation. Your reasons may include that you accept the risk or will define compensating controls.

A waiver may apply to the current user, constraint, or constraint within a dimension such as the business unit.

Resolving Conflicts

The risk tolerance of the enterprise determines whether a segregation of duties conflict must be removed from the security reference implementation.

The following approaches resolve conflicts.

• Change the segregation of duties policy.
• Ensure a job role does not contain incompatible duties.
• Define data security policies that restrict authorized access by incompatible duties.

Changing a segregation of duties policy may not be possible in most cases. For example, a policy that segregates creation of payables invoice from making payables payments should be preserved, even if the Accounts Payables Manager job role includes a duty role for each activity. To prevent an accounts payables manager from being authorized to perform both duties, or from being authorized to make payables payments to self and direct reports, the Accounts Payables Manager job role must be changed. The security implementation can be changed to include two job roles that segregate the incompatible duties. Added data security policy grants can restrict the access to at risk data.
For information on managing segregation of duties, see the Oracle Application Access Controls Governor Implementation Guide and Oracle Application Access Controls Governor User's Guide.

Role Provisioning and Segregation of Duties: How They Work Together

Segregation of duties (SOD) checks occur when roles are assigned to users. The checks are based on Oracle Application Access Controls Governor (AACG) policies. The Oracle Identity Management (OIM) integration includes predefined routing rules for remediation in the Manage IT Security business process.

External users such as suppliers or partners need to be provisioned with roles to facilitate access to parent company interfaces and data. The process by which such provisioning requests are approved in Oracle Fusion Applications helps explain the request flows and possible outcomes.

Note

In Oracle Identity Management (OIM), external users means users who are not specific to applications, such as enterprise roles or the absence of entitlement to access an application.

The figure shows the role provisioning request flow. OIM uses AACG to check segregation of duties violations.
A supplier or partner requests admission to a program using an implementation of the Supplier Portal Submission. The submission is captured in one or both of the following tables in advance of approving or rejecting the supplier or partner.

- Oracle Fusion Trading Community Model
- Interface Staging

Oracle Fusion Applications collects the employee names for the supplier or partner company at the time the company submits its request to join the program so that all employees accessing Oracle Fusion Applications on behalf of the supplier or partner are provisioned.

AACG in the Oracle Governance, Risk and Compliance Controls (GRCC) suite is certified to synchronize with the policy and identity stores for all pillars or partitions of Oracle Fusion Applications and integrated with the Oracle Fusion Applications security approach to roll up entitlements (by means of duty roles) to the roles that are provisioned to internal users. SOD policies can be defined and enforced at any level of authorization. For external users, SOD policies use attribute information stored in the Trading Community Model tables.

### OIM and the SPML Client

Enterprise business logic may qualify the requester and initiate a role provisioning request by invoking the Services Provisioning Markup Language (SPML) client module, as may occur during onboarding of internal users with Human Capital Management (HCM), in which case the SPML client submits an asynchronous SPML call to OIM. Or OIM handles the role request by presenting roles for selection based on associated policies.

OIM recognizes the role provisioning request and initiates a call to AACG.

OIM apprises the SPML client of the current state of the role provisioning request as SOD_CHECK_IN_PROGRESS.

OIM stores the SOD check result as part of OIM audit data.

OIM apprises SPML client of the current state of the SPML request. The provisioning is either still in progress with segregation of duties being checked, or conflicts were found. If conflicts exist, AACG rejects the request and notifies the application.

<table>
<thead>
<tr>
<th>Status</th>
<th>Conflicts</th>
<th>Current State</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOD_CHECK_IN_PROGRESS</td>
<td>Unknown</td>
<td>Request sent to AACG and waiting for response</td>
</tr>
<tr>
<td>SOD_REMEDIATION_IN_PROGRESS</td>
<td>Conflict found</td>
<td>AACG detected violations and remediation is in progress</td>
</tr>
<tr>
<td>SOD_CHECK_APPROVED</td>
<td>No conflict found</td>
<td>No SOD violations found</td>
</tr>
<tr>
<td>SOD_CHECK_REJECTED</td>
<td>Conflict found</td>
<td>AACG detected violations that cannot be remediated</td>
</tr>
<tr>
<td>SOD_REMEDIATION_APPROVED</td>
<td>Conflict found</td>
<td>AACG detected violations that are approved</td>
</tr>
</tbody>
</table>
In the absence of an SOD exception, OIM provisions all relevant users.

<table>
<thead>
<tr>
<th>SOD_REMEDIATION_REJECTED</th>
<th>Conflict found</th>
<th>AACG detected violations that are rejected by approver</th>
</tr>
</thead>
</table>

Note

When a partner user is provisioned, all employees of the partner enterprise are provisioned. SOD checks occur when an external user requests to join a program, because SOD policies operate across Oracle Fusion Applications, not at the individual level. Supplier or partner company user requests are not approved if there is an SOD conflict against the supplier company.

OIM provides AACG with the details of SOD exception approval workflow. AACG audits the outcome for use in future detective controls and audit processes.

**Oracle Application Access Controls Governor**

AACG may respond with the following.

- Roles may be provisioned to the external user or its employees because no SOD conflict is found
- SOD conflict is found and request is denied because the relevant SOD policy is to be strictly enforced and no exception approval should be allowed
- SOD conflict is found and the exception to the policy is allowed, so the request goes through additional processing, such as an approval process.

Supplier or Partner Relationship Management responds to an SOD exception by updating Trading Community Model tables with the current state. An enterprise may elect to implement a landing pad that offers external users a means of addressing the SOD problem by providing more information or withdrawing the request.

SOD violation checking occurs during role implementation and provisioning, and can be turned on or off if AACG is provisioned and enabled as part of the Oracle Fusion Applications deployment.

**Segregation of Duties Exception Resolution or Approval Workflow**

Depending upon status, OIM kicks off an auditable SOD exception resolution workflow. Resolution can be conditional based on approval or requirements such as contracts being met.

If one of the paths for exception resolution is to get an approval, then the SOD exception resolution drives the approval using AMX. Standard AMX rules, not business rules, resolve the approval for the SOD exception, including the following.

- Organizational hierarchies
- Multiple mandatory and optional approvers
- Rerouting and approval delegation

The approver resolution uses AMX Rules Designer to access various user attributes and organizational hierarchies managed in Oracle Fusion Applications repositories. This information is typically not available in OIM or the LDAP identity store repository. Enterprises can define additional approval rules using AMX Thin Client.

The SOD Exception Approver gets a notification through supported channels that a new request is awaiting approval. The approver signs in to the global SOA federated worklist application that aggregates all pending worklist items for the user from all Oracle Fusion applications and logical partitions or pillars of applications. The SOD exception approval tasks show up in the same list.

The SOD exception approval task shows the details of the SPML request and SOD Provisioning results in a page rendered by OIM. The approver may take one of the following actions.

- Approve the request as it is
- Reject the request

If the approver approves the request, OIM sends an SOD_REMEDICATION_APPROVED status to the SPML client.

If the approver rejects the request, OIM sends an SOD_REMEDICATION_REJECTED status to the SPML client. The provisioning request is considered completed with a failure outcome and the external users is notified. Oracle Fusion Applications updates the Trading Community Model tables with the rejected status.

**Remediation Task Assignments**

The SOD remediation tasks are assigned based on the role being requested.

1. If the role requested is Chief Financial Officer, the SOD remediation task is assigned to the IT Security Manager role.

2. If the SOD violation results from a policy where the SOD control tag is the Information Technology Management business process and the control priority is 1, the SOD remediation task is assigned to Application Administrator role.

3. In all other scenarios, the SOD remediation task is assigned to the Controller role.

For more information about configuring audit policies, see the Oracle Fusion Applications Administrator's Guide.

For information on managing segregation of duties, see the Oracle Application Access Controls Governor Implementation Guide and Oracle Application Access Controls Governor User's Guide.
Common Applications Configuration: Define Help Configuration

Define Help Configuration: Overview

The Define Help Configuration task list contains tasks that let you set up and maintain Oracle Fusion Applications Help for all users. Use the Set Help Options task to determine if certain aspects of Oracle Fusion Applications Help are available to users and to control how aspects of the help site work. Use the Assign Help Text Administration Duty and Manage Help Security Groups tasks to set up customization of help content.

After performing the help configuration tasks, you can review the predefined help and consider whether to add or customize any content. You can also customize help that is embedded in the application, for example hints and help windows, using other tools such as Oracle JDeveloper and Oracle Composer.

Use the Setup and Maintenance work area to access the tasks in the Define Help Configuration task list.

Set Help Options

Help Feature Choices and Help Options: Points to Consider

Help feature choices on the Configure Offerings page in the Setup and Maintenance work area control the look and behavior of Oracle Fusion Applications Help, and also determine which help options are available. Help options are setup options on the Set Help Options page.

Local Installation of Help

Select the Local Installation of Help feature choice so that the Define Help Configuration task list appears in your implementation project, and you can select two additional features (Access to Internet-Based Help Features and Help Customization) to control the fields available on the Set Help Options page.

Access to Internet-Based Help Features

Select this feature choice to provide users access to features that involve navigation to sites on the Web. If you select this feature choice, then the Web Sites Available from Help Site section is available on the Set Help Options page.
Important

Some help includes links to the Oracle Fusion Applications Technology Library. If you select this feature, then these links open the library on the Oracle Technology Network Web site. If you do not select this feature, then you must apply the Technology Library patch from My Oracle Support so that users can access the library locally.

Help Customization

Select the Help Customization feature choice if you intend to customize predefined help or add your own files to help. For example, you can add internal policies or procedures as help, and Oracle User Productivity Kit content, if any. Not selecting this feature choice only hides the relevant setup options; help customization is not disabled for users with customization access. To disable help customization, remove the Oracle Fusion Help Text Administration duty from the job roles where it is predefined.

If you select this feature choice, then the Custom Help Security feature choice is available, as well as all these sections on the Set Help Options page:

- Custom Help
- User Productivity Kit
- Privacy Statement

Custom Help Security

Select this feature choice if you want certain help files to be available only to a restricted set of users. You can define the user groups allowed to view corresponding help files. Do not select this feature choice if you do not have this requirement, because the feature can have an impact on performance.

If you select the Custom Help Security feature choice, then the Manage Help Security Groups task is available in the Define Help Configuration task list in your implementation project. There are no help options associated with this feature choice.

Administering Collaboration Features and Announcements in Help: Points to Consider

Announcements and collaboration features (discussions, ratings and comments) allow users to share information regarding help and the subjects that particular help files cover. These collaboration features are also used elsewhere in Oracle Fusion Applications. Announcements and discussions may not be available in a public cloud delivery of Oracle Fusion Applications.

Use the Set Help Options page in the Setup and Maintenance work area to enable the announcements and discussions features and to set options about ratings. When administering these features, consider the purpose of each feature and points that are specific to Oracle Fusion Applications Help.
Announcements

Use announcements to broadcast information to all users of your help site. You can provide information about help, for example new custom help that was recently added, or about anything that users should take note of, for example a change in company policy. Announcements can appear on any of the tabs on the home page of Oracle Fusion Applications Help. You can target specific user groups by posting announcements to specific tabs, for example, posting information related to implementation to the Functional Setup tab.

Only administrators for discussions can post announcements to the help site. For more information on granting administrator roles for discussions, see the Oracle Fusion Middleware Administrator’s Guide for Oracle WebCenter.

Discussions

Users can use discussions to post questions or comments about subjects covered in specific help files. For example, after reading help on expense reports, users might have questions or comments about company policies or processes for expenses. Other users who later access this help file would benefit from the information in the discussion.

You can set a help option to enable discussions. Each help file would contain a Discuss link that all users can use to read discussions about that file. They can also start a discussion topic or post to existing topics. These discussions are visible only to users in your enterprise.

---

**Important**

Do not enable discussions until servers for discussions are up and running.

---

Ratings and Comments

Users can rate any help file on a five star system and provide feedback about the content. This information is helpful to other users in deciding which help file to open. Help files with a higher average rating are listed first in help windows, and in the help listings you see as you browse using the help navigators.

The scope of ratings and reviews is limited to your enterprise.

---

**FAQs for Set Help Options**

**When do I link to the Oracle User Productivity Kit library from the help site?**

Provide a link to your Oracle User Productivity Kit (UPK) library if you have UPK licensed and custom UPK content to share with your users. You give them access to a library of custom UPK content in addition to any custom UPK demos that you added to the help site itself. UPK demos that you add as custom help are available only in the See It mode, so the library can include the same demo in other modes. If you have UPK versions earlier than 3.6.1, then you cannot add UPK demos as custom help, so the link is the only way for users to access custom UPK content from the help site.
How can I find the URL to the Oracle User Productivity Kit library?

The URL to enter on the Set Help Options page should be the full path from the Web server where you are hosting your Oracle User Productivity Kit (UPK) content to the index.html file that opens the table of contents for the library, for example, http://<your domain>.com/UPKcontent/PlayerPackage/index.html.

In this example, you or your UPK administrator would publish one UPK player package that contains all the content to be linked to from Oracle Fusion Applications Help, as well as the index.html file, and place the PlayerPackage folder in a manually created folder called UPKcontent on the Web server.

FAQs for Assign Help Text Administration Duty

Who can add and manage custom help?

Users with the Oracle Fusion Help Text Administration duty role have access to customize help in Oracle Fusion Applications Help. This duty is assigned by default to various job roles, in particular the administrators for product families.

You can assign the duty role to other users who need access to customize help. Use the Manage Duties task in the Setup and Maintenance work area to search for the Oracle Fusion Help Text Administration duty role on the Role Catalog page, and map additional job roles to this duty role.

Manage Help Security Groups

Creating Help Security Groups: Worked Example

This example demonstrates how to create a help security group to define a set of job roles that have access to help. The help security group can then be assigned to particular help files so that only users with any of the defined roles have access to the help.

The following table summarizes key decisions for this scenario.

<table>
<thead>
<tr>
<th>Decisions to Consider</th>
<th>In This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>What type of users do you need to limit help access to?</td>
<td>Human resources (HR) specialists</td>
</tr>
<tr>
<td>Is there a specific time period for which this access is needed?</td>
<td>No, the help files should always be viewed only by the HR specialists</td>
</tr>
<tr>
<td>Where do you want this group to appear in the list of values for help security groups?</td>
<td>First</td>
</tr>
</tbody>
</table>

Define a help security group and assign a duty role to the group.

1. From the Setup and Maintenance work area, find the Manage Help Security Groups task and click Go to Task.

3. Complete the fields, as shown in this table. Leave the start and end dates blank.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help Security Group</td>
<td>HR</td>
</tr>
<tr>
<td>Meaning</td>
<td>HR Only</td>
</tr>
<tr>
<td>Description</td>
<td>Viewing by HR specialists only</td>
</tr>
<tr>
<td>Display Sequence</td>
<td>1</td>
</tr>
</tbody>
</table>

4. Click **Save**.

5. With your new help security group selected, go to the Associated Roles section and add a new row.

6. Select **PER_HUMAN_RESOURCE_SPECIALIST** as the role name.

7. Click **Save and Close**.

You have created a new lookup code for the Help Security Groups lookup type, which is a standard lookup. The lookup code has the name, meaning, and description that you defined for the help security group.

You have also created a data security policy for the help database resource, specifying that the Human Resource Specialist role can view help that is defined with the HR security group. If you go to the Manage Database Resources and Policies page and find the database resource, or object, ATK_KR_TOPICS, then you can see the policy for the Human Resource Specialist role, with the condition that the column name, SECURITY_CODE, is equal to the value HR.

---

**Help File Customization**

**Help File Customization: Overview**

If you have the appropriate job roles, then you can customize the help files in the help site. Use the Manage Custom Help page to maintain both predefined and custom help files. You can create, duplicate, edit, and delete custom files, or set their status to Active or Inactive. For predefined files, you can only duplicate them or set their status. For each help file, predefined or custom, use help locations to determine where the help file appears in the application and in the help site. You have various options in how you add custom help, for example by uploading a file or specifying a URL.

**Note**

To make a copy of all custom help for testing, migration, or other purposes, use the export and import feature in the Setup and Maintenance work area.

There are various ways to access help customization.
• Many help files can be accessed from help windows in the application. If you want to customize help in the context of a help window, for example create a custom help file and add a link to it from a specific help window, then start by opening that help window. When you click the Manage Custom Help link, you go to the Manage Custom Help page, and the help location fields are automatically populated with values that correspond to the help window. This way you can easily select existing files to add to the same help location, and when you create a new file, the same help location appears by default.

• Open the Manage Custom Help page directly from the home page of Oracle Fusion Applications Help or from search result pages.

• To edit a specific file, you can either find it in the Manage Custom Help page, or open the file itself and click the Edit link.

• Likewise, you can find glossary terms in the Manage Custom Help page, or click the Glossary link in the global area to open the Glossary tab, search for the term, and click Edit.

Note
When you search in the Manage Custom Help page, make sure that the Custom Help Only check box is not selected if you are looking for predefined help.

Help Types: Explained

Oracle Fusion Applications Help contains various types of help content, including demos, examples, FAQs, glossary terms, help topics, and PDF guides. A business process or product can be supported by some or all of these help types.

Demo

Demos are Oracle User Productivity Kit (UPK) topics that visually demonstrate how to use the application to complete a short task or portion of a task. Demos can also provide an introduction to complex dashboards and work areas.

Example

Examples provide real use cases of features to illustrate how and when to use the feature, or scenarios to illustrate abstract concepts. Worked examples show exactly what you need to do to achieve a specific result, emphasizing decisions that you make and values that you enter.

FAQ

FAQs, or frequently asked questions, provide brief answers to questions that you might have regarding a task or page. For example, they can briefly explain what a term means, why something happened, how you can perform an action, or what happens if you perform the action.
Glossary

Glossary terms provide definitions for words or phrases used in help. You can search or browse glossary terms in the Glossary tab of Oracle Fusion Applications Help. Where the links are available, you can also see the definition when you hover over the term in help content for other help types.

Help Topic

Help topics explain key concepts, illustrate how application components work together, or assist in decision-making by explaining points to consider or the options you have. Help topics can also provide reference, overview, and other information.

PDF Guide

PDF guides present a collection of help content from the other help types, except demos, in an organized and logical format. For example, there are guides addressing specific business processes and setup offerings. You can see lists of all guides from the Guides menu in Oracle Fusion Applications Help.

Help Locations: Explained

Help locations determine where users can find help files, custom or not, from either the application or the help site.

Help locations include:

- Page or section values
- Help hierarchies
- Primary locations

Page or Section Values

The value in the Page or Section field on the help customization pages represents where users can click a help icon to open a help window that contains a link to the help file. In most cases, this value represents a page or region header in the application. Help windows are also available on specific tabs or windows, and in the Setup and Maintenance work area for specific task lists or tasks. You can associate a help file with multiple page or section values, or with none at all.

The page or section value reflects the logical navigation to the help window. For example, Edit Opportunity page, Revenue tab, Recommendations window does not mean that the help file is available in three different places. The help icon is in the Recommendations window, which is accessed from the Revenue tab on the Edit Opportunity page.

If the value suggests multiple locations, for example Create and Edit Opportunity pages, then the help file is available from the page header of both the Create Opportunity and Edit Opportunity pages. If the page or section value is, for example, a dashboard region that appears in multiple dashboards, then
the value does not specify the page name but just the region. The help file is available from that region in multiple dashboards.

**Help Hierarchies**

Help files are associated with help hierarchies, which are used to categorize help files and aid users in finding help. Each help file can have multiple hierarchies, with at least one of type Business Processes. The business process hierarchy is based on the Business Process Management model. Every page or section value is predefined with a specific business process hierarchy. If you select a page or section without entering a business process hierarchy, the predefined hierarchy appears by default.

The Search by Business Process navigator in the help site is based on the business process hierarchy. For example, if you assign two business process hierarchies to a help file, users can find the file in both locations in the navigator. When the user clicks More Help from a help window, all help files assigned to the same business process hierarchy as the page or section value are returned as search results.

Similarly, the Search by Product navigator is based on the Product hierarchy type, in which level 1 is the product family, level 2 is the product, and level 3 is the business activity owned by that product.

The Search by Functional Setup navigator is based on the Functional Setup hierarchy type. The level 1 nodes for this hierarchy are:

- Functional Setup Manager, which includes help about using the Setup and Maintenance work area.

- Guides, which contains level 2 nodes that correspond to business areas and setup offerings. All the user reference and functional setup PDF guides are included.

- Offerings, which contains level 2 nodes for each setup offering, and lower levels for the main task lists in the offerings. Help for the task lists and tasks are included.

The Search by Common Tasks navigator is based on the Welcome hierarchy type. The level 1 nodes represent categories of functional areas common to all users.

**Primary Locations**

The primary location of a help file designates the hierarchy that is displayed for the help file in search results and within the help content as breadcrumbs. You cannot change the primary location of a help file that came with your help installation. Primary locations of predefined help are based on the business process hierarchy, while custom help files can have primary locations based on hierarchies of any type.

**Editing Predefined Help and Glossary Terms: Points to Consider**

When you open any predefined help file, including glossary terms, that came with Oracle Fusion Applications Help, you can see an edit option if you have roles allowing edit access. When you edit predefined help, keep in mind:
• What happens to the original help file
• Where predefined help appears
• Considerations specific to glossary terms

What Happens to the Original Files

When you edit predefined help, you are actually creating a new custom help file based on the original file, with the same help locations. The customized version replaces the original, which becomes inactive and hidden from users. You can display both versions by reactivating the original in the Manage Custom Help page.

Note

In the Search Results: Existing Help region on the Manage Custom Help page, there is no option to edit predefined help. You can duplicate a predefined help file, edit the copy, and optionally inactivate the original.

Where Predefined Help Appears

All predefined help comes with preassigned help locations, including at least one based on the hierarchy of type Business Processes. Many also have predefined page or section values that indicate where the help can be accessed from help windows in the application.

To change where predefined help appears, either in the help site navigators or in the application, create a duplicate in the Manage Custom Help page. Change or add help locations to your custom copy, and inactivate the original.

Even though glossary terms do not appear in the help site navigators, you still need to enter at least one help location to categorize the glossary term.

Considerations Specific to Glossary Terms

When you edit a predefined glossary term, the original term becomes inactive. Existing links to the glossary term, from other predefined and custom help files, will automatically point to your custom version. If you later inactivate the custom glossary term, make sure to activate the original term so that the links still work.

Links in Custom Help: Points to Consider

When you create or edit custom help, follow best practices when you include links to help files or other content. If you are working on custom help created by duplicating a predefined help file, then you may see existing links from the original file in the Help Content section. The types of links that you can work with include:

• Related help links
• Standard hypertext links
• Links to documentation library content
• Glossary term links

For all link types, except the standard hypertext links, you must create or edit custom help with a Text or Desktop source type. In other words, you must type the help content directly in the application or use an HTML file that you upload to help. For standard hypertext links, the source type can also be URL.

Related Help Links

Related help is the section at the end of help files that contains links to other help files. The syntax for related help contains a comma-separated list of IDs that represent help files.

This figure provides an example of related links code.

You can delete this code to remove all related help, or delete individual links (for example, CREATE_AUTOMATIC_POSTING_CRITERIA_S_0000).

Restriction

Aside from deleting related links to predefined help, do not edit them in any other way.

Standard Hypertext Links

You can create standard hypertext links to any file or Web site as long as you ensure the stability and validity of the links, including links to other help files, custom or not. These links can appear anywhere in the body of your help file as long as they come before any related help links.

In the Help Content section, highlight the text that you want to use as link text and click the Add Link icon button.

For links to other help files, open the file to which you want to link, and click the E-Mail link. Use the URL in the autogenerated e-mail text as the link to the file.

Links to Documentation Library Content

The syntax for links to HTML files in documentation libraries, for example the Oracle Fusion Applications Technology Library, is:

WCSUG4636 is the anchor ID and Understanding Tags is the link text. You can:

• Modify the link by replacing the existing anchor ID or editing the link text, or both.
• Remove the link by deleting all the code for it.
• Create links to documentation library content by following the same syntax. These links can appear anywhere in the body of your help file as long as they come before any related help links.

Important
To ensure that you are linking to a supported documentation library, enter anchor IDs only from documentation libraries that are linked from predefined help topics.

Glossary Term Links

Glossary term links provide definitions in a note box when users hover over the term in help files.

This figure shows an example of code for a glossary term link.

\[\text{Glossary Term:} \text{accounting period}, \text{ACCOUNTING\_PERIOD\_001} \]

In this example, \textit{accounting period} is the link text, or glossary term, and ACCOUNTING\_PERIOD\_001 is the identifier, or title ID.

- To remove the link but retain the text, delete all the code except the term itself.

- To add custom glossary term links, you must follow the link syntax and use the correct title ID for the glossary term.

To find the title ID for a custom glossary term, search for it in the Manage Custom Help page or the Glossary tab to access the Edit Help page for the term. The title ID is displayed on the Edit Help page.

Note

If your custom help has glossary terms and the source type is Desktop File, then make sure before uploading that the quotes around the glossary term are actual quotation marks in raw HTML, not \&quot. Otherwise, quotation marks will appear when users view the help file.

Customizing PDF Guides: Worked Example

This example demonstrates how to customize a PDF guide that came with Oracle Fusion Applications Help. This guide is currently not available from any help window in the application.

The following table summarizes key decisions for this scenario.

<table>
<thead>
<tr>
<th>Decisions to Consider</th>
<th>In This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>What changes do you need to make to the guide?</td>
<td>Change the title of a chapter and remove a section in that chapter, to hide content about a particular subject</td>
</tr>
<tr>
<td>Which help window should the customized guide appear in?</td>
<td>The help window for the entire Welcome dashboard of Oracle Fusion Applications</td>
</tr>
</tbody>
</table>
Which help navigators should the customized guide appear in, and on which node? | Same as the original guide, plus the path associated with the help window
---|---
Do you want to limit access to the customized guide? | No, same as the original guide

Edit a copy of the original PDF guide, and use the Manage Custom Help page to replace the original PDF guide with your new file.

**Copying and Editing the PDF Guide**

1. Open the original PDF guide from the help site and save a copy to your desktop. Leave open the help file for the guide.
2. Using a PDF editor application, change the title of the chapter wherever the chapter title appears. Delete the content you want to hide from users.
3. Make sure that your new PDF guide is less than 6 MB.

**Replacing the Original PDF Guide**

1. In the help file that you still have open for the original PDF guide, click the Edit link.
2. On the Create Help page, use the default values except where indicated.
3. Update the title to the name that you want to display to users.
4. In the File Name field, browse for and select your customized guide.
5. Delete any keywords or parts of the description relevant to the content you removed from the PDF guide.
6. Add a help location with the Business Processes hierarchy type and select Information Technology Management as the level 1 node, Manage Enterprise Application Software as the level 2 node, and Use Applications as the level 3 node.
7. Select Welcome page in the Page or Section column.
8. Click Save and Close. The help file for the original PDF guide is automatically set to inactive.

**Adding Custom UPK Content to Help: Worked Example**

This example demonstrates how to add custom Oracle User Productivity Kit (UPK) topics as demo help files. These help files function like any predefined help file for demos. You can search and include these files in help windows and navigators as you would other help.

In this scenario, you are adding two demos about social networking, to appear in help windows on the Welcome dashboard.

**Note**

Your demo must be made with UPK 3.6.1 or later to be added as help.

The following table summarizes key decisions for this scenario.
Decisions to Consider | In This Example
--- | ---
What UPK content do you want to add to help? | From a UPK module containing five topics, add two as custom demos on the help site

Which help navigators should each demo appear in, and on which node? | Because the two demos are about social networking:
- Search by Common Tasks navigator, under the Collaboration node
- Search by Business Process navigator, under Information Technology Management - Manage Networking and Communications - Manage Social Networking Capabilities

Which help window should each demo appear in? | On the Welcome dashboard of Oracle Fusion Applications, one demo goes in the help window in the Activity Stream region, and the other in the People Connection region

Do you want to limit access to the help files for the demos? | No

Do you want the help files to appear in the New and Updated pane? | Yes

Generate a report of UPK document IDs, which you will use when creating custom help, to identify the UPK topics that you want to add. Publish the UPK module as a player package, then create custom help for the UPK topics that you want to use as help demos.

**Generating a UPK Document ID Report**

1. In the UPK Developer, select **Details View**.
2. Right-click any column header, for example Name, and select **Column Chooser**.
3. In the Column Chooser dialog box, click and drag the Document ID column header and drop it after the Name column. Close the Column Chooser dialog box.
4. From the File menu, select to print, and save the output as a Microsoft Excel file to your desktop.

**Creating the Player Package**

1. From the UPK Developer, make sure that the topics that you want to add as demos have the See It play mode. The topics can also have other modes, but only the See It mode is included in the custom help file.
2. Publish the module, specifying any location for the output and selecting to publish the selection only.
3. In the Formats section of the Publish Content window, select the **Player** check box under the **Deployment** check box group.
4. In the Player section, select the **Include HTML Web Site** check box, to ensure that the custom help file includes a text-only version of the UPK topic.
5. Finish the publishing process, after optionally setting other options.
6. Navigate to the location where you specified the output to be generated.
7. In the Publishing Content folder, copy the PlayerPackage folder and add it to the web server where you store UPK content.

Creating Custom Help for Demos

1. Open the help window in the Activity Stream region on the Welcome dashboard of Oracle Fusion Applications, and click Manage Custom Help.

2. On the Manage Custom Help page, the page or section and hierarchy values are populated with the values for the Activity Stream region.

3. Click Create.

4. On the Create Help page, complete the fields in the General Information section, as shown in this table. Use the default values except where indicated.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>The name of the UPK topic.</td>
</tr>
<tr>
<td>Source Type</td>
<td>Oracle User Productivity Kit</td>
</tr>
<tr>
<td>File Location</td>
<td>The full URL of the player package folder on the Web server, for example, http://&lt;your domain&gt;.com/UPKcontent/PlayerPackage.</td>
</tr>
<tr>
<td>Document ID</td>
<td>The document ID of the UPK topic to add to the help window in the Activity Stream region. You can copy and paste this ID from the Microsoft Excel file that you generated earlier.</td>
</tr>
<tr>
<td>Help Type</td>
<td>Demo</td>
</tr>
<tr>
<td>Help Security Group</td>
<td>Unsecured</td>
</tr>
<tr>
<td>Keywords</td>
<td>Terms relevant to the demo.</td>
</tr>
<tr>
<td>Description</td>
<td>Summary of the demo.</td>
</tr>
<tr>
<td>Include in New and Updated pane</td>
<td>Selected</td>
</tr>
</tbody>
</table>

The Help Location section contains values for the help window in the Activity Stream region. This help file will also appear in the Search by Business Process navigator under this predefined hierarchy.

5. Click Save and Close.

6. On the Manage Custom Help page, open the help locations for the help file that you just created.

7. Add a help location with the Welcome hierarchy type and select Collaboration Features as the level 1 node.

8. Add another help location with the Business Processes hierarchy type and select Information Technology Management as the level 1 node, Manage Networking and Communications as the level 2 node, and Manage Social Networking Capabilities as the level 3 node.

9. Click Save and Close.
10. Starting at the Connections region, repeat steps 1 to 9 for the other UPK topic that you want to add.

**FAQs for Help File Customization**

**How can I restrict help content to specific user roles?**

When you create or edit help, select a help security group that represents the set of roles that you want to have access to the help. If you do not see the Security Group field, then your administrator has not selected the Custom Help Security feature choice. The Unsecured group has no associated roles, so anyone can view the help. The predefined Secured group includes all internal employees and contingent workers, unless this group has been edited. You can create security groups and associate roles using the Manage Help Security Groups page, which you can access by starting in the Setup and Maintenance Overview page and searching for the Manage Help Security Groups task. Your new security groups are immediately available for use to secure new or edited help files.

**Why can't I select and add help to a location?**

You must specify a page or section to add the existing help to. To ensure that help is added to the correct help window, go to the page or section in the application, click the Help icon, and click the Manage Custom Help link in the help window. Alternatively, in the Manage Custom Help page, search for at least a page or section and a level 1 value for the Business Processes hierarchy type before selecting the Select and Add option.

You cannot select and add help to a particular hierarchy, on the Manage Custom Help page, without a page or section. To add just a hierarchy, search for the help file, add a new help location, and specify only the hierarchy information.

**What happens to custom help when a help patch is applied?**

Oracle Fusion Applications Help patches update all help files, both active and inactive, except custom help. Custom help files are not affected by patches. Consider reviewing inactive files to see if you want to activate the updated version, or to make similar edits to the custom versions of those files, if any.

**Embedded Help Customization**

**Customizing Embedded Help: Highlights**

You can customize help that is embedded in the application, for example hints and help windows, for all users of Oracle Fusion Applications.

Embedded help customization is fully described in the Oracle Fusion Applications Extensibility Guide.

- Edit, create, or delete hint text that appears on hover over buttons, links, icons, or tab titles.
See: Customizing or Adding Bubble Embedded Help

• Edit, create, or delete other types of embedded help.

See: Customizing or Adding Static Instructions, In-field Notes, Terminology Definitions, and Help Windows
Common Applications Configuration: Define Source Systems

Source Systems: Explained

Source systems are used to import data into Oracle Fusion Applications, and are used within the application to identify source data information. You can specify whether the source system is a Spoke system, such as a legacy system, or a Purchased system, such as data from a third party provider. You can also specify what type of data will be imported using the source system, for example, you can specify that a source system will import trading community members.

You can configure the following for a source system:

- Source system code, name, and description
- Source system type
- Enable for Items, Trading Community Members, Order Orchestration and Planning, and Assets

**Source System Code, Name, and Description**

You can create a source system code to uniquely identify the source system. Source system codes are used by the application when creating references between source IDs and the Oracle Fusion Applications database IDs. You can create a source system name and description to provide information that is more descriptive than the source system code.

**Note**

You cannot update the source system code once you have created the source system.

**Source System Type**

You must set up a source system as either a Spoke system, such as a legacy system, or a Purchased system, such as data from Dun & Bradstreet.
Enable for Items, Trading Community Members, Order Orchestration and Planning, and Assets

You should select which types of entities will be imported from the source system into the Oracle Fusion Applications database from the following:

- Items
- Trading Community Members
- Order Orchestration and Planning
- Assets

You can select one or more of these entity types as required for the source system. It is important to enable the correct entity types because each import UI filters source systems based on their entity type. For example, if a source system is enabled for Trading Community Members, Items, and Assets, then the source system can be selected as a data source in the Trading Community Members, Items, and Asset import UIs, but the source system cannot be selected in the Orchestration and Planning import UI.

Source System Entities: Explained

Source System Entities are the entities, such as addresses and parties, which can be imported using a specified source system.

When you import data from a source system, all of the entities in the source system data will be imported. Within the Source System Entities UI, you can choose to allow multiple source references, which allows multiple records from a source system to map to a single trading community record.

FAQs for Define Source Systems

What happens if I allow multiple source system references?

Allowing multiple source system references means that when you import data from a source system you can merge multiple, or duplicate, source system records and create one record in the Oracle Fusion Applications database.

If you do not allow multiple source system references then an Oracle Fusion Applications database record will be created for every source system record. This means that you could potentially create duplicate records in the Oracle Fusion Applications database.
Common Applications Configuration: Define Application Toolkit Configuration

Define Application Toolkit Configuration: Overview

Oracle Fusion Application Toolkit (ATK) is an application that provides various core components of Oracle Fusion Applications, including the Welcome dashboard, Oracle Fusion Applications Help, the Reports and Analytics pane, and the Watchlist feature. Use the Define Application Toolkit Configuration task list to set up and maintain some of these components for all users, and the Define Help Configuration task list for Oracle Fusion Applications Help.

Use the Setup and Maintenance work area to access the tasks in the Define Application Toolkit Configuration task list.

Map Reports to Work Areas

Additional Report Setup in the Context of the Reports and Analytics Pane: Highlights

Aside from determining which work areas a specific report is mapped to, you can perform additional setup for reports in the context of the Reports and Analytics pane. You can set up report permissions, and enable Oracle Business Intelligence (BI) Publisher reports for scheduled submission.

This additional setup is described in the Oracle Fusion Middleware User’s Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition) and the Oracle Fusion Applications Extensibility Guide.

Report Permissions

- You can restrict access to specific reports for specific users, and this security is not limited to the Reports and Analytics pane. Refer to the Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition).

See: Assigning Permissions
Oracle Business Intelligence Publisher Reports Submission

- Oracle BI Publisher reports must be registered as processes with Oracle Enterprise Scheduler to be enabled for scheduling. This registration also enables a Schedule link for the report in the Reports and Analytics Pane. Refer to the Oracle Fusion Applications Extensibility Guide, and perform the following steps in the specified order.

  - Create an Oracle Enterprise Scheduler job definition for the report.
    
    See: Tasks Required to Run Custom Reports with Oracle Enterprise Scheduler Service

  - Specify the job definition details in the report’s properties.
    
    See: Enabling Reports for Scheduling from the Reports and Analytics Pane

FAQs for Map Reports to Work Areas

How can I set up the Reports and Analytics pane for all users?

You can remove any currently mapped report from the Reports and Analytics pane, or add mappings to reports from the Oracle Business Intelligence (BI) Presentation catalog. To access the setup, click Edit Settings in the Reports and Analytics pane, or use the Map Reports to Work Areas task in the Setup and Maintenance work area. If you do the former, then you can set up only the Reports and Analytics pane on the work area that you are in.

If you do the latter, then you can select a work area to set up. If you do not see the desired work area, most likely you do not have access to it due to security. You can request to be granted a role that has access to the work area, or another administrator or business user with access to the work area can be granted the Reports and Analytics Region Administration Duty to be able to map reports to the work area.

Tip

On the Map Reports to Work Areas page only, you can also use the Synchronize button to remove mappings to reports that are no longer in the catalog, for all work areas at once.

Any changes you make in either UI apply to all users with access to the mapped work area.

Why can’t I see reports when mapping reports to work areas for the Reports and Analytics pane?

It is possible that there are no reports currently mapped to the work area that you select in the Map Reports to Work Areas page. Alternatively, reports are mapped, but you do not see them due to security.

Similarly, in the list of all available reports from the catalog, you can see only the reports that you have access to. You can request to be granted a role that has
access to the reports that you want to map, or another administrator or business user with access to those reports can be granted the Reports and Analytics Region Administration Duty to be able to map reports to work areas.

**Why can’t I see reports when I edit settings for the Reports and Analytics pane?**

In the Edit Settings window, you may not be able to see a currently mapped report because you do not have access to it due to security.

Similarly, in the list of all available reports from the catalog, you can see only the reports that you have access to. You can request to be granted a role that has access to the reports that you want to map, or another administrator or business user with access to those reports can be granted the Reports and Analytics Region Administration Duty to be able to map reports to work areas.

---

**Set Watchlist Options**

**Watchlist Setup: Points to Consider**

For all users across the site, you can disable or enable predefined Watchlist categories and items, edit their names, and determine how often item counts refresh. You cannot delete predefined Watchlist categories and items, nor create any for the site. Users can create their own Watchlist items through saved searches.

Access the Set Watchlist Options page by starting in the Setup and Maintenance Overview page and searching for the Set Watchlist Options task.

**Disabling Predefined Categories and Items**

Use the Set Watchlist Options page to enable or disable predefined Watchlist categories and items. Disabling any category or item also disables associated processes involved in calculating the Watchlist item counts for all users. These processes include creating data caches, performing security checks, invoking services across domains, running queries, and so on.

An item with the **Predefined** type represents the actual predefined Watchlist item that appears in the Watchlist. If you disable this type of Watchlist item, then:

- The item is not available for users to display in their watchlist
- The item is removed from any watchlist where it is currently displayed

A Watchlist item with the **User-created saved search** type does not appear in the Watchlist; it controls the display of the **Manage Watchlist** button or menu item in pages with saved searches. If you disable this type of Watchlist item, then:

- The **Manage Watchlist** option is not available to users in the corresponding work area, so users cannot use their own saved searches as Watchlist items. A message is displayed to users when they try to use this option.
• Any user-defined saved searches from that work area already used as Watchlist items are no longer available in the users’ watchlist. The user-defined saved searches are still available to be used for searching, but not for the Watchlist.

If you disable a Watchlist category, then the category is not available for users to include in their watchlist, and all Watchlist items within the category are also disabled.

Note

If a product is not installed, then corresponding roles should not be granted to users, so the categories for that product would not be displayed. Even if roles are granted, users would see only zero count items. To make sure that only appropriate categories are available, disable categories for all products that are not installed, and disable irrelevant items, if any, for products that are installed.

Ultimately, the Watchlist for any user contains the subset of categories and items that are enabled in the Set Watchlist Options page:

• Plus any items based on user-defined saved searches
• Minus any categories or items that the user chooses to hide using Watchlist preferences
• Minus any items with no results found, if the user chooses to hide such items using Watchlist preferences

Specifying Refresh Intervals

All Watchlist items have a predefined refresh interval, which controls how often the query that calculates the count for a Watchlist item can be run. Use the Set Watchlist Options page to edit the interval values. What you specify as the refresh interval for a Watchlist item of type User-created Saved Search applies to all Watchlist items based on saved searches created by users on the corresponding search page.

When the user is in the Welcome dashboard with the Watchlist open for at least two and a half minutes, the query automatically runs for all Watchlist items if no refresh already ran in this user session. To subsequently run the query again, users can manually refresh the Watchlist region. The Refresh icon is enabled after five minutes since the last refresh.

Note

During a refresh, the query runs for an individual Watchlist item only if the time since the last query for this item is equal to or greater than the specified refresh interval. Since the manual refresh of the entire Watchlist is not available until five minutes after the last refresh, you should not set a Watchlist item refresh interval that is less than five minutes.

When users open Watchlist from the global area, a refresh automatically runs if five minutes have passed since the last refresh. During this refresh, the query
runs for an individual Watchlist item only if the time since the last query for this item is equal to or greater than the specified refresh interval.

For example, you set the interval to eight minutes for a particular Watchlist item. When the user signs in and goes to the Welcome dashboard, with the Watchlist open, the query automatically runs for this Watchlist item after two and a half minutes. Every two and a half minutes after, a check is performed for stale counts and new cached counts are displayed.

Five minutes after the query ran, the Refresh icon is enabled and the user performs a manual refresh. However, the query does not run for this Watchlist item, because the refresh interval is eight minutes. The user navigates away from the Welcome dashboard and opens the Watchlist from the global area six minutes later. A refresh automatically runs because more than five minutes have passed since the last refresh. This time, the query runs for this Watchlist item because it has been more than eight minutes since the query last ran for this item.

**Editing Predefined Category and Item Names**

Predefined Watchlist category and item names are stored as meanings of standard lookups. Lookup types for predefined categories end with WATCHLIST, for example EXM_EXPENSES_WATCHLIST. Edit the lookup type meaning to change the category name. To change item names, edit lookup code meanings for that lookup type.
Common Applications Configuration: Maintain Common Reference Objects

Maintain Common Reference Objects: Overview

The Maintain Common Reference Objects task list contains Oracle Fusion Middleware Extensions for Applications (Applications Core) tasks that support implementation of common behaviors, such as data security or reference data sets.

Use this task list to manage common reference objects that are defined centrally and shared across applications, in addition to those that are specific to Applications Core functionality. You can access this task list by starting in the Setup and Maintenance Overview page and searching for common reference object task lists.

Note

Offerings also include application-specific tasks for managing Applications Core objects. For example, the Financials offering includes tasks such as Manage Receivables Descriptive Flexfields, and Manage Receivables Lookups.

For more information on configuring custom objects, see the Oracle Fusion Applications Extensibility Guide.

To make the Maintain Common Reference Objects task list available in your implementation project, go to Setup and Maintenance Overview - Configure Offerings, and for a specific offering, select the Maintain Common Reference Objects feature choice.

Define Application Taxonomy

Application Taxonomy: Highlights

Application taxonomy is the organization of Oracle application components and functions in a hierarchical structure, from product lines to logical business areas. This hierarchy represents a breakdown of products into units based on how applications are installed and supported. Maintain this hierarchy on the Manage
A detailed introduction to application taxonomy is provided in the Oracle Fusion Applications Developer’s Guide.

Hierarchy

- The application taxonomy hierarchy contains various levels and types of nodes, or modules.

  See: Characteristics of the Level Categories

  See: Benefits of a Logical Hierarchy

Usage

- Use application taxonomy to understand relationships among applications and between an application and its files. This information is helpful in managing various phases of the product lifecycle.

  See: How to Manage the Lifecycle

Modules in Application Taxonomy: Explained

A module is any node in the application taxonomy hierarchy. The top level of the hierarchy is product line, followed by product family, application, and logical business area. There can be multiple levels of logical business areas, with one or more nested within a parent logical business area.

Product Line

A product line is a collection of products under a single brand name, for example, Oracle Fusion.

Product Family

A product family is a collection of products associated with a functional area that may or may not be licensed together as a single unit, for example Financials.

Application

An application is a single product within a product family, containing closely related features for a specific business solution, for example General Ledger.

Logical Business Area

A logical business area is a collection of business object definitions organized into a logical grouping. It contains the model objects, services, and UI components for those business objects. Logical business areas have their own hierarchy levels and in some cases can be two or three levels deep. Each leaf node has at least one business object and service, up to a maximum of four business objects and associated services. A logical business area with more than four business objects are further refined with child logical business area levels.
Each of these parent-child levels is represented by a directory in the physical package hierarchy.

Managing Modules in Application Taxonomy: Points to Consider

Manage modules on the Create Child Module or Edit Module page, which you can access by starting in the Setup and Maintenance Overview page and searching for the Manage Taxonomy Hierarchy task. When you create a module, it is a child of the currently selected node in the application taxonomy hierarchy. This determines which values are available, for example for module type. Once created, you cannot delete the module or move it elsewhere in the hierarchy. As you create or edit modules, consider the following points regarding specific fields.

Identifiers

Module ID is the unique primary key for nodes in the taxonomy table. When you create a module, an ID is automatically generated. Once the module is created, you cannot update the ID.

Module key and alternative ID are additional identifiers of the module, presented in a way that is easier to read than the module ID. The module key is a string identifier, for example AP for the Oracle Fusion Payables application. The alternative ID is a numeric identifier, for example 1 for the Oracle Fusion product line. These IDs are provided for the product line, product family, and application modules, but you can optionally add them for logical business areas and new custom modules.

Note

Do not change the module key or alternative ID for predefined modules.

The product code is relevant only to application and logical business area modules. You can leave the field blank for other module types. The product code for applications is the short name that can be displayed in lists of application values, for example FND for Oracle Fusion Middleware Extensions for Applications.

Names

Module name is the logical name for the module and is always available. The name must be unique among nodes in the same hierarchy level with the same parent, but try to make it as unique in the whole hierarchy as possible.

The user name and description can appear to users in other parts of Oracle Fusion Applications, so make sure that the values are something that users know to represent the module.

Usage Types

Though you can update the usage type to reflect the current state of the module, just doing so does not affect the actual state. For example, setting a module as installed does not mean it is actually installed if the installation itself has not taken place. Installation refers to operations related to laying down all the
components needed to create an Oracle Fusion Applications environment, while deployment is the process that starts the managed servers and clusters and facilitates the actual use of product offerings. A licensed module is available for installation and deployment, and a deployed module is considered actively used when actually used by users.

**Seed Data**

If seed data is allowed, then seed data such as flexfields and lookups can be extracted for the module using seed data loaders. By default, extract is allowed for all predefined modules of type application and logical business area.

**Associations**

You can associate a logical domain to modules of type product family, as well as one or more enterprise applications to modules of type application. This association represents the relationship between the taxonomy modules and the corresponding domain and enterprise applications stored in the Oracle Fusion Applications Functional Core (ASK) tables.

**Define Reference Data Sharing**

Reference data sharing facilitates sharing of configuration data such as jobs and payment terms, across organizational divisions or business units. You define reference data sets and determine how the data is shared or partitioned. Use reference data sets to reduce duplication and maintenance by sharing common data across business entities where appropriate. Depending on the requirement (specific or common), each business unit can maintain its data at a central location, using a set of values either specific to it or shared by other business units.

You can share reference data after it is filtered on the basis of sets. A common reference data set is available as the default set, which can be assigned to several business units sharing the same reference data. For commonly used data such as currencies, you can use the common reference data set and assign it to multiple business units in various countries that use the same currency. In cases where the default set cannot be assigned to an entity, you can create specific sets. The data set visible on the transactional page depends on the sharing method used to share reference data.

For example, XYZ Corporation uses the same grades throughout the entire organization. Instead of managers in different business units setting up the same grades, XYZ Corporation decides to create a set called Grades and assign the grades reference data group for all business units in the organization to the Grades set, so that the grades can be shared.

**Note**

For specific information on configuring reference data sharing for a particular object or product, refer to its product documentation.
Reference Data Sets: Explained

Reference data sets are logical groups of reference data that can be accessed by various transactional entities depending on the business context. Oracle Fusion Applications contains a common reference data set as well as an enterprise set that may be used as a default set. Depending on your business requirement you can create and maintain additional reference data sets, while continuing to use the common reference data set.

Consider the following scenario.

Your enterprise can decide that some aspects of corporate policy should affect all business units and leave other aspects to the discretion of the business unit manager. This allows your enterprise to balance autonomy and control for each business unit. For example, if your enterprise holds business unit managers accountable for their profit and loss, but manages working capital requirements at a corporate level, you can let managers define their own sales methods, but define payment terms centrally. In this case, each business unit would have its own reference data set for sales methods, and there would be one central reference data set for payment terms assigned to all business units.

Partitioning

The partitioning of reference data and creation of data sets enable you to create reference entities across tables or lookup types, and share modular information and data processing options among business units. With the help of partitioning, you can choose to create separate sets and subsets for each business unit depending upon its business requirement, or create common sets or subsets to enable sharing reference data between several business units, without the need for duplicating the reference data. Partitioning provides you the flexibility to handle the reference data in a way appropriate to your business needs.

The following figure illustrates the reference data sharing method (assignment to one set only, with common values) where the user can access the data assigned to a specific set in a particular business unit, as well as access the data assigned to the common set.
Reference Data Sets and Sharing Methods: Explained

Oracle Fusion Applications reference data sharing feature is also known as SetID. The reference data sharing functionality supports operations in multiple ledgers, business units, and warehouses, thereby reducing the administrative burden and decreasing the time needed to implement new business units. For example, you can share sales methods, transaction types, or payment terms across business units or selected other data across asset books, cost organizations, or project units.

The reference data sharing features use reference data sets to which reference data is assigned. The reference data sets group assigned reference data. The sets can be understood as buckets of reference data assigned to multiple business units or other application components.

Reference Data Sets

You begin this part of your implementation by creating and assigning reference data to sets. Make changes carefully as changes to a particular set will affect all business units or application components using that set. You can assign a separate set to each business unit for the type of object that is being shared. For example, assign separate sets for payment terms, transaction types, and sales methods to your business units.

Your enterprise can decide that some aspects of corporate policy should affect all business units and leave other aspects to the discretion of the business unit manager. This allows your enterprise to balance autonomy and control for each business unit. For example, if your enterprise holds business unit managers accountable for their profit and loss, but manages working capital requirements at a corporate level, you can let managers define their own sales methods, but define payment terms centrally. In this case, each business unit would have its own reference data set for sales methods, and there would be one central reference data set for payment terms assigned to all business units.

The reference data sharing is especially valuable for lowering the cost of setting up new business units. For example, your enterprise operates in the hospitality industry. You are adding a new business unit to track your new spa services. The hospitality divisional reference data set can be assigned to the new business unit to quickly setup data for this entity component. You can establish other business unit reference data in a business unit specific reference data set as needed.

Reference Data Sharing Methods

There are variations in the methods used to share data in reference data sets across different types of objects. The following list identifies the methods:

- Assignment to one set only, no common values allowed. The simplest form of sharing reference data that allows assigning a reference data object instance to one and only one set. For example, Asset Prorate Conventions are defined and assigned to only one reference data set. This set can be shared across multiple asset books, but all the values are contained only in this one set.

- Assignment to one set only, with common values. The most commonly used method of sharing reference data that allows defining reference data
object instance across all sets. For example, Receivables Transaction Types are assigned to a common set that is available to all the business units without the need to be explicitly assigned the transaction types to each business unit. In addition, you can assign a business unit specific set of transaction types. At transaction entry, the list of values for transaction types includes transaction types from the set assigned to the business unit, as well as transaction types assigned to the common set that is shared across all business units.

- Assignment to multiple sets, no common values allowed. The method of sharing reference data that allows a reference data object instance to be assigned to multiple sets. For instance, Payables Payment Terms use this method. It means that each payment term can be assigned to one or more than one set. For example, you assign the payment term Net 30 to several sets, but the payment term Net 15 is assigned to only your corporate business unit specific set. At transaction entry, the list of values for payment terms consists of only one set of data; the set that is assigned to the transaction’s business unit.

Note: Oracle Fusion Applications contains a reference data set called Enterprise. Define any reference data that affects your entire enterprise in this set.

### Assigning Reference Data Sets to Reference Objects: Points to Consider

You can assign the reference data sets to reference objects on the Manage Reference Data Set Assignments page. For multiple assignments, you can classify different types of reference data sets into groups and assign them to reference entity objects. The assignment takes into consideration the determinant type, determinant, and reference group, if any.

#### Determinant Types

The partitioned reference data is shared based on a business context setting called the determinant type. It is the point of reference used in the data assignment process. The following table lists the determinant types used in the reference data assignment.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset Book</td>
<td>Information about the acquisition, depreciation, and retirement of an asset that belongs to a ledger or a business unit.</td>
</tr>
<tr>
<td>Business Unit</td>
<td>The departments or organizations within an enterprise.</td>
</tr>
<tr>
<td>Cost Organization</td>
<td>The organization used for cost accounting and reporting on various inventory and cost centers within an enterprise.</td>
</tr>
<tr>
<td>Project Unit</td>
<td>A logical organization within an enterprise that is responsible for enforcing consistent project management practices.</td>
</tr>
<tr>
<td>Reference Data Set</td>
<td>References to other shared reference data sets.</td>
</tr>
</tbody>
</table>
Determinant

The determinant or determinant value is the value that corresponds to the selected determinant type. The determinant is one of the criteria for selecting the appropriate reference data set. For example, when managing set assignments for the set determinant type, Reference Data Set is the determinant type, and you would enter the corresponding set code value as the corresponding determinant value.

Reference Groups

A transactional entity may have multiple reference entities (generally considered to be setup data) that are treated in the same manner because of commonness in implementing business policies and legal rules. Such reference entities in your application are grouped into logical units called reference groups, based on the functional area and the partitioning requirements that they have in common. For example, all tables and views that define Sales Order Type details might be part of the same reference group.

Note

The reference groups are predefined in the reference groups table and are available for selection and assignment.

Define Lookups

Lookups: Explained

Lookups are lists of values in applications. You define a list of values as a lookup type consisting of a set of lookup codes, each code’s translated meaning, and optionally a tag. End users see the list of translated meanings as the available values for an object.

Lookups provide a means of validation and lists of values where valid values appear on a list with no duplicate values. For example, an application might store the values Y and N in a column in a table, but when displaying those values in the user interface, Yes or No (or their translated equivalents) should be available for end users to select. For example, the two lookup codes Y and N are defined in the REQUIRED_INDICATOR lookup type.

In another example, a lookup type for marital status has lookup codes for users to specify married, single, or available legal partnerships.

<table>
<thead>
<tr>
<th>Lookup Type</th>
<th>Lookup Code</th>
<th>Meaning</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAR_STATUS</td>
<td>M</td>
<td>Married</td>
<td></td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>Single</td>
<td></td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>Registered Partner</td>
<td>+NL</td>
</tr>
<tr>
<td></td>
<td>DP</td>
<td>Domestic Partner</td>
<td>-FR, AU</td>
</tr>
</tbody>
</table>
In this case, tags are used for localizing the codes. All legislations list Married and Single. Only the Dutch legislation lists Registered Partner. And all legislations except France and Australia also list Domestic Partner.

When managing lookups, you need to understand the following:

- Using lookups in applications
- Customization levels
- Accessing lookups
- Enabling lookups
- The three kinds of lookups: standard, common, and set enabled

**Using Lookups in Applications**

Use lookups to provide validation or a list of values for a user input field in a user interface.

An example of a lookup used for validation is a flexfield segment using a table-validated value set with values from a lookup type. An example of a lookup in a list of values is a profile option's available values from which users select one to set the profile option. Invoice Approval Status gives the option of including payables invoices of different approval statuses in a report. The lookup code values include All so that users can report by all statuses: Approved, Resubmitted for approval, Pending or rejected, and Rejected.

**Customization Level**

The customization level of a lookup type determines whether the lookups in that lookup type can be edited. This applies data security to lookups.

Some lookup types are locked so no new codes and other changes can be added during implementation or later, as needed. Depending on the customization level of a lookup type, you may be able to change the codes or their meanings. Some lookups are designated as extensible, so new lookup codes can be created during implementation, but the meanings of predefined lookup codes cannot be modified. Some predefined lookup codes can be changed during implementation or later, as needed.

The customization levels are user, extensible, and system. The following table shows which lookup management tasks are allowed at each customization level.

<table>
<thead>
<tr>
<th>Allowed Task</th>
<th>User</th>
<th>Extensible</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deleting a lookup type</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Inserting new codes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Updating start date, end date, and enabled fields</td>
<td>Yes</td>
<td>Yes, only if the code is not predefined data</td>
<td>No</td>
</tr>
<tr>
<td>Deleting codes</td>
<td>Yes</td>
<td>Yes, only if the code is not predefined data</td>
<td>No</td>
</tr>
<tr>
<td>Updating tags</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Updating module</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Predefined data means LAST_UPDATED_BY = SEED_DATA_FROM_APPLICATION.

If a product depends on a lookup, the customization level should be system or extensible to prevent deletion.

Once the customization level is set for a lookup type, it cannot be modified. The customization level for lookup types created using the Define Lookups page is by default set at the User level.

**Standard, Common, and Set-Enabled Lookups**

The available kinds of lookups are as follows.

<table>
<thead>
<tr>
<th>Lookup</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>Lists the available codes and translated meanings</td>
</tr>
<tr>
<td>Set enabled</td>
<td>Additionally associates a reference data set with the lookup codes</td>
</tr>
<tr>
<td>Common</td>
<td>Legacy lookups</td>
</tr>
</tbody>
</table>

Standard lookups are the simplest form of lookup types consisting only of codes and their translated meaning. They differ from common lookups only in being defined in the standard lookup view.

Common lookups exist for reasons of backward compatibility and differ from standard lookups only in being defined in the common lookup view.

Set enabled lookup types store lookup codes that are enabled for reference data sharing. At runtime, a set-enabled lookup code is visible because the value of the determinant identifies a reference data set in which the lookup code is present.

**Accessing Lookups**

Standard, set-enabled, and common lookups are defined in the Standard, Set-enabled, and Common views, respectively. Applications development may define lookups in an application view to restrict the UI pages where they may appear.

In lookups management tasks, lookups may be associated with a module in the application taxonomy to provide a criteria for narrowing a search or limiting the number of lookups accessed by a product specific task such as Manage Purchasing Lookups.

**Enabling Lookups**

A lookup type is reusable for attributes stored in multiple tables.

Enable lookups based on the following.

- Selecting an **Enabled** check box
• Specifying an enabled start date, end date, or both
• Specifying a reference data set determinant

If you make changes to a lookup, users must sign out and back in before the changes take effect. When defining a list of values for display rather than validation, limit the number of enabled lookup codes to a usable length.

Managing a Standard Lookup: Example

Creating a new standard lookup involves creating or selecting a lookup type to which the lookup code belongs, and determining appropriate values for the lookup codes and their meanings.

Note

You can only create or edit the lookup codes for a particular lookup type if its customization level supports it.

Creating a Lookup Type Called COLORS

Your enterprise needs a list of values for status to be used on various objects such as processes or users. The lookups are colors, so the lookup type you create is COLORS.

<table>
<thead>
<tr>
<th>Lookup type parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup type name</td>
<td>COLORS</td>
</tr>
<tr>
<td>Meaning</td>
<td>Status</td>
</tr>
<tr>
<td>Description</td>
<td>Status by color</td>
</tr>
<tr>
<td>Module</td>
<td>Oracle Fusion Middleware Extensions for Applications</td>
</tr>
</tbody>
</table>

The lookup codes you define for the COLORS lookup type are, BLUE, RED, GREEN, and YELLOW.

<table>
<thead>
<tr>
<th>Lookup Code</th>
<th>Meaning</th>
<th>Enabled</th>
<th>Display Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLUE</td>
<td>Urgent</td>
<td>No</td>
<td>4</td>
</tr>
<tr>
<td>RED</td>
<td>Stop</td>
<td>Yes</td>
<td>1</td>
</tr>
<tr>
<td>GREEN</td>
<td>Go</td>
<td>Yes</td>
<td>3</td>
</tr>
<tr>
<td>YELLOW</td>
<td>Caution</td>
<td>Yes</td>
<td>2</td>
</tr>
</tbody>
</table>

Understanding the Resulting Data Entry List of Values

Users need to respond to a process question by indicating whether to stop it, use caution, go ahead, or complete it urgently.

The list of values for the COLORS lookup type includes the meanings for the enabled codes.
Analysis

The BLUE lookup code was not enabled and does not appear in the list of values. The display sequence of values in the list of values is alphabetical unless you enter a number manually to determine the order of appearance. Number 1 indicates the value listed first in the list of values.

Note

Only lookups that are enabled and active, meaning between start and end dates, are visible.

When users enter one of the values from the list of values for the lookup type COLORS, the transaction table records the lookup code. In this example, the code is stored in the Status column.

<table>
<thead>
<tr>
<th>Transaction number</th>
<th>User name</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jane</td>
<td>RED</td>
</tr>
<tr>
<td>2</td>
<td>Bob</td>
<td>YELLOW</td>
</tr>
<tr>
<td>3</td>
<td>Alice</td>
<td>BLUE</td>
</tr>
</tbody>
</table>

The status for one user is BLUE because at the time they entered a value, BLUE was enabled. Disabling a lookup code does not affect transaction records in which that code is stored. Data querying and reporting have access to disabled lookup codes in transaction tables.

Managing Set-Enabled Lookups: Examples

Creating a new set-enabled lookup is similar to creating a standard lookup with the addition of specifying a reference data set determinant for the lookup codes.

Note

You can only create or edit the lookup codes for a particular lookup type if its customization level supports it.

The reference data set for a set-enabled lookup code is part of its foreign key. This is unlike other set-enabled entities.

Selecting a Reference Group for a Set-Enabled Lookup Type

By specifying a reference group for a set-enabled lookup type you indicate which reference data set assignments are available for its lookup codes. For example a COLORS lookup type might be set enabled for a Countries reference group that includes the US and EU reference data set assignments.
Selecting a Reference Data Set for a Set-Enabled Lookup

The reference data set determines which lookup code is included in the list of values. If a COLORS lookup type contains a RED, YELLOW, ORANGE, and GREEN lookup code, you can enable one RED lookup as coming from the US reference data set and another RED lookup as coming from the EU reference data set with divergent meanings.

<table>
<thead>
<tr>
<th>Reference Data Set</th>
<th>Lookup Code</th>
<th>Lookup Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>RED</td>
<td>Red</td>
</tr>
<tr>
<td>US</td>
<td>YELLOW</td>
<td>Yellow</td>
</tr>
<tr>
<td>US</td>
<td>GREEN</td>
<td>Green</td>
</tr>
<tr>
<td>EU</td>
<td>RED</td>
<td>Rouge</td>
</tr>
<tr>
<td>EU</td>
<td>ORANGE</td>
<td>Orange</td>
</tr>
</tbody>
</table>

In addition to divergent meanings for lookup codes based on associated reference data set, some lookup codes may be unique to one or another reference data set as the ORANGE lookup is to the EU reference data set in this example.

In another example, a lookup type called HOLD_REASON provides a list of reasons for applying a hold to a contract renewal. Reference data sets determine which codes are included in the hold reason list of values.

<table>
<thead>
<tr>
<th>Reference Data Set</th>
<th>Lookup Code</th>
<th>Lookup Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>SEC</td>
<td>SEC Compliance Review</td>
</tr>
<tr>
<td>US</td>
<td>DIR</td>
<td>Needs Director’s Approval</td>
</tr>
<tr>
<td>US</td>
<td>VP</td>
<td>Needs Vice President’s Approval</td>
</tr>
<tr>
<td>CHINA</td>
<td>CSRC</td>
<td>Pending China Securities Regulatory Commission Review</td>
</tr>
<tr>
<td>CHINA</td>
<td>PR</td>
<td>Needs President’s Approval</td>
</tr>
<tr>
<td>COMMON</td>
<td>REQUESTED</td>
<td>Customer Request</td>
</tr>
</tbody>
</table>

Using the Manage Set Assignments task, you have defined assignments that designate the China business unit to refer to the CHINA and the US business unit to refer to the US and all business units to refer to the COMMON set. When end users place a contract hold in the US business unit, only the three reason codes in US_SET are available. When placing a contract hold in the China business, only the two codes in China_SET are available.

FAQs for Define Lookups

How can I edit lookups?

You can edit the existing lookup codes of a lookup type or add new lookup codes on the Define Lookups pages, which you can access by starting in the Setup and Maintenance work area and searching for lookup tasks. You can edit the existing lookup codes of a lookup type, or add new lookup codes to a lookup type, if the customization level for the lookup type supports editing.
Why can't I see my lookup types?

Lookups are listed by lookup type. Typically lookup types are managed using tasks that handle a group of related lookups, such as Manage Geography Lookups. Each task gives you access only to certain lookup types. The generic tasks provide access to all lookups types of a kind, such as all common lookups using the Manage Common Lookups task.

If existing lookups are not available to the tasks of the Define Lookups activity, they may be validated for use in a lookup view that is not central to all applications or whose owning application has not been specified in a lookup view.

Lookups can only be managed in the Define Lookups tasks if the lookup’s view application is the standard lookups view, common lookups view, or set-enabled lookups view. Lookups defined in an application view can only be managed by following instructions provided by the owning application.

Note

A lookup type and its codes can only be defined in one lookup view.

What's the difference between a lookup type and a value set?

A lookup type consists of lookup codes that are the values in a static list of values. Lookup code validation is a one to one match.

A table-validated value set can consist of values that are validated through a SQL statement, which allows the list of values to be dynamic.

Tip

A table validated value set can be defined based on any table, including the lookups table. This allows a lookup type to be made into a table-validated value set that can be used in flexfields.

<table>
<thead>
<tr>
<th>Area of Difference</th>
<th>Lookup Type</th>
<th>Value Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of values</td>
<td>Static</td>
<td>Dynamic if Table validation type</td>
</tr>
<tr>
<td>Validation of values</td>
<td>One to one match of meaning to code included in a lookup view, or through the determinant of a reference data set</td>
<td>By format or inclusion in a table</td>
</tr>
<tr>
<td>Format type of values</td>
<td>char</td>
<td>varchar2, number, and so on</td>
</tr>
<tr>
<td>Length of value</td>
<td>Text string up to 30 characters</td>
<td>Any type of variable length from 1 to 4000</td>
</tr>
<tr>
<td>Duplication of values</td>
<td>Never. Values are unique.</td>
<td>Duplicate values allowed</td>
</tr>
<tr>
<td>Management</td>
<td>Managed by both administrators and end-users, except system lookups or predefined lookups at the system customization level, which cannot be modified.</td>
<td>Maintained by administrators, except some product flexfield codes, such as GL for Oracle Fusion General Ledger, which are maintained by end users</td>
</tr>
</tbody>
</table>
A lookup type cannot make use of a value from a value set.

Value sets can make use of standard, common, or set-enabled lookups.

Both lookup types and value sets are used to create lists of values from which users select values.

**What's a lookup tag used for?**

Tags on lookup codes allow you to add a label to your lookup codes.

Lookup tags are unvalidated and uninterpreted by lookups. A tag can be used to categorize lookups based on facilitating searches or guiding how a lookup should be used.

Document what the tag on a lookup represents and how to use it.

---

**Manage Messages**

**Messages: Highlights**

The message dictionary contains messages that tell users about business rule errors, such as missing or incorrect data, and how to resolve them, to warn users about the consequences of intended actions, and provide information in log files. These messages are defined for specific applications and modules, but a few are common messages that can be used in any application. All applications also use messages stored outside of the message dictionary.


**Managing Messages**

- Use the Manage Messages page to create and edit custom messages in the message dictionary, as well as edit predefined messages. Do not delete predefined messages unless you are sure that they are not used anywhere. Refer to the Oracle Fusion Applications Developer's Guide.

  See: Introduction to Message Dictionary Messages

- Messages outside of the message dictionary, for example confirmations and field validations, are managed in Oracle Application Development Framework Faces components or through message resource bundles used for translation. Refer to the Oracle Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework.

  See: Displaying Hints and Error Messages for Validation and Conversion
Creating and Editing Messages: Highlights

Each message in the message dictionary has many attributes and components, including message properties, text, and tokens, that you define when creating or editing the message.

Details about these messages are described in the Oracle Fusion Applications Developer’s Guide.

**Message Properties**

- The message type identifies the type of information that the message contains.
  
  See: Understanding Message Types

- The message name and number are identifiers for the message. There are specific message number ranges for predefined messages in each application, and you should not edit numbers assigned to predefined messages. When creating custom messages, use only message numbers within the 10,000,000 to 10,999,999 range.

  See: About Message Names

  See: About Message Numbers

- The translation notes for predefined messages might contain internal content that you can disregard.

  See: About Translation Notes

- The message category, severity, and logging enabled option are related to the incident and logging process.

  See: About Grouping Messages by Category and Severity

  See: Understanding Incidents and Diagnostic Logs with Message Dictionary

**Message Text and Tokens**

- The message text comprises various components, some of which are displayed only to select users. To determine which component of the message text is displayed to a particular user, set the Message Mode profile option (FND_MESSAGE_MODE) at the user level for that user. The message component short text is visible to all users and therefore, the profile option does not apply to this component. Also, the profile option applies only to messages in the message dictionary.

  See: About Message Components

- Tokens are variables that represent values to be displayed in the message text.

  See: About Tokens
Common Messages: Points to Consider

Common messages, which have message names that begin with FND_CMN and message numbers between 0 and 999, are used throughout Oracle Fusion Applications. Each common message can appear in multiple places in any product family. For example, the FND_CMN_NEW_SRCH message can be used for any search to indicate that no results were found. Common messages that are of type error or warning are part of the message dictionary.

Editing Common Messages

Because a common message can be used in any application, consider the ramifications if you edit any aspect of the message, including incident and logging settings. Changes would be reflected in all instances where the message is used. For example, if you change the message text, make sure that the text would make sense to all users across Oracle Fusion Applications who might see it.

Creating Common Messages

You can create custom common messages for use in multiple places within a single product. Do not begin the message name with FND_CMN, but use another suitable convention. The message number should be within the range that is designated for the product.

Define ISO Reference Data

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.

**Natural Languages: Points to Consider**

Natural languages are all the languages that humans use, written and spoken. If a language is enabled, then users can associate it with entities, for example as languages spoken by sales representatives. When managing natural languages, consider tasks to perform and best practices for entering particular values.

**Tasks**

Once you add a language, it cannot be deleted, just disabled. You can optionally associate natural languages with International Organization for Standardization (ISO) languages and territories, just for reference.

**Values**

When you create a natural language, use the alpha-2 ISO code as the language code, or, if not available, then alpha-3. If the language is not an ISO language, then use x- as a prefix for the code, for example x-ja for a Japanese dialect.
Use the sgn code of ISO-639-2 for sign languages, followed by territory code, for example sgn-US for American Sign Language. You can also use Internet Assigned Numbers Authority (IANA) language tags.

The natural language description should be the language name with territory name in parenthesis where needed, for example English (Australia) and English (Canada).

**FAQs for Define ISO Reference Data**

**When do I create or edit territories?**

Edit territory descriptions to determine how they are displayed in lists of country values throughout Oracle Fusion Applications. The predefined territories are all countries from the International Organization for Standardization (ISO) 3166 standard. You usually would not edit territory names or codes.

Do not edit National Language Support (NLS) territory codes, which are identifiers used in the system, unless you need to change the association between ISO and system territory. You usually would not edit the default currency, which is the value that defaults in the Currency field in Oracle Fusion Applications user preferences after the user first selects a territory.

Create territories if new countries emerge and the system has not yet been patched with the latest ISO country values.

**When do I create or edit industries?**

Edit industry descriptions to determine how they are displayed in Oracle Fusion Applications. You usually would not edit industry names, which are from the North American Industry Classification System (NAICS). Enabled industries are mainly used in the context of customization, though these values can also appear in any application.

Create industries if you have particular ones you need, for example for customization, that are not included in the NAICS standard.

**When do I associate industries with territories?**

Optionally associate industries with territories to provide an industry in territory value, used for customization. For example, administrators can customize a page in one way for users within an industry in one country, and another way for users within the same industry in another country. The administrator would select the appropriate industry in territory value to set the customization context.

**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.

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.

When do I create or edit ISO languages?

You can edit the names and descriptions of International Organization for Standardization (ISO) languages to determine how they are displayed in lists of ISO language values in Oracle Fusion Applications. The ISO languages are from the ISO 639 standard. If there were changes to the ISO standard and the system has not yet been patched with the latest ISO values, you can update the ISO alpha-2 code or add languages as needed.

When do I edit languages?

Installed languages automatically appear on the Manage Languages page, so you do not manually enter newly installed languages. This page contains all languages available for installation and translation in Oracle Fusion Applications. Each dialect is treated as a separate language. The language codes and names are values used by the system.

You generally would not edit any of the detailed fields unless you really need to and know what they are.

When do I create or edit time zones?

Though all standard time zones are provided, optionally enable only a subset for use in lists of time zone values in Oracle Fusion Applications. You can add time
zones if new zones became standard and the system has not yet been patched with the latest values.

Manage Data Security Policies

Data Security in the Security Reference Implementation: Explained

The reference implementation contains a set of data security policies that can be inspected and confirmed to be suitable or a basis for further implementation using the Authorization Policy Manager (APM).

The security implementation of an enterprise is likely a subset of the reference implementation, with the enterprise specifics of duty roles, data security policies, and HCM security profiles provided by the enterprise.

The business objects registered as secure in the reference implementation are database tables and views.

Granting or revoking object entitlement to a particular user or group of users on an object instance or set of instances extends the base Oracle Fusion Applications security reference implementation without requiring customization of the applications that access the data.

Data Security Policies in the Security Reference Implementation

The data security policies in the reference implementation entitle the grantee (a role) to access instance sets of data based on SQL predicates in a WHERE clause.

---

**Tip**

When extending the reference implementation with additional data security policies, identify instance sets of data representing the business objects that need to be secured, rather than specific instances or all instances of the business objects.

---

Predefined data security policies are stored in the data security policy store, managed in the Authorization Policy Manager (APM), and described in the Oracle Fusion Applications Security Reference Manual for each offering. A data security policy for a duty role describes an entitlement granted to any job role that includes that duty role.

---

**Warning**

Review but do not modify HCM data security policies in APM except as a custom implementation. Use the HCM Manage Data Role And Security Profiles task to generate the necessary data security policies and data roles.

---

The reference implementation only enforces a portion of the data security policies in business intelligence that is considered most critical to risk management without negatively affecting performance. For performance reasons it is not practical to secure every level in every dimension. Your enterprise
may have a different risk tolerance than assumed by the security reference implementation.

HCM Security Profiles in the Security Reference Implementation

The security reference implementation includes some predefined HCM security profiles for initial usability. For example, a predefined HCM security profile allows line managers to see the people that report to them.

The IT security manager uses HCM security profiles to define the sets of HCM data that can be accessed by the roles that are provisioned to users.

Data Roles

The security reference implementation includes no predefined data roles to ensure a fully secured initial Oracle Fusion Applications environment.

The security reference implementation includes data role templates that you can use to generate a set of data roles with entitlement to perform predefined business functions within data dimensions such as business unit. Oracle Fusion Payables invoicing and expense management are examples of predefined business functions. Accounts Payable Manager - US is a data role you might generate from a predefined data role template for payables invoicing if you set up a business unit called US.

HCM provides a mechanism for generating HCM related data roles.

Data Security: Explained

By default, users are denied access to all data.

Data security makes data available to users by the following means.

- Policies that define grants available through provisioned roles
- Policies defined in application code

You secure data by provisioning roles that provide the necessary access. Enterprise roles provide access to data through data security policies defined for the inherited application roles.

When setting up the enterprise with structures such as business units, data roles are automatically generated that inherit job roles based on data role templates. Data roles also can be generated based on HCM security profiles. Data role templates and HCM security profiles enable defining the instance sets specified in data security policies.

When you provision a job role to a user, the job role implicitly limits data access based on the data security policies of the inherited duty roles. When you provision a data role to a user, the data role explicitly limits the data access of the inherited job role to a dimension of data.

Data security consists of privileges conditionally granted to a role and used to control access to the data. A privilege is a single, real world action on a
single business object. A data security policy is a grant of a set of privileges to a principal on an object or attribute group for a given condition. A grant authorizes a role, the grantee, to actions on a set of database resources. A database resource is an object, object instance, or object instance set. An entitlement is one or more allowable actions applied to a set of database resources.

Data is secured by the following means.

<table>
<thead>
<tr>
<th>Data security feature</th>
<th>Does what?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data security policy</td>
<td>Grants access to roles by means of entitlement</td>
</tr>
<tr>
<td>Role</td>
<td>Applies data security policies with conditions to users through role provisioning.</td>
</tr>
<tr>
<td>Data role template</td>
<td>Defines the data roles generated based on enterprise setup of data dimensions such as business unit.</td>
</tr>
<tr>
<td>HCM security profile</td>
<td>Defines data security conditions on instances of object types such as person records, positions, and document types without requiring users to enter SQL code</td>
</tr>
<tr>
<td>Masking</td>
<td>Hides private data on non-production database instances</td>
</tr>
<tr>
<td>Encryption</td>
<td>Scrambles data to prevent users without decryption authorization from reading secured data</td>
</tr>
</tbody>
</table>

The sets of data that a user can access via roles are defined in Oracle Fusion Data Security. Oracle Fusion Data Security integrates with Oracle Platform Security Services (OPSS) to entitle users or roles (which are stored externally) with access to data. Users are granted access through the entitlement assigned to the roles or role hierarchy with which the user is provisioned. Conditions are WHERE clauses that specify access within a particular dimension, such as by business unit to which the user is authorized.

**Data Security Policies**

Data security policies articulate the security requirement "Who can do What on Which set of data," where 'Which set of data' is an entire object or an object instance or object instance set and 'What' is the object entitlement.

For example, accounts payable managers can view AP disbursements for their business unit.

<table>
<thead>
<tr>
<th>Who</th>
<th>do</th>
<th>what</th>
<th>on which set of data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounts payable managers</td>
<td>view</td>
<td>AP disbursements</td>
<td>for their business unit</td>
</tr>
</tbody>
</table>

A data security policy is a statement in a natural language, such as English, that typically defines the grant by which a role secures business objects. The grant records the following:

- Table or view
- Entitlement (actions expressed by privileges)
Instance set (data identified by the condition)

For example, disbursement is a business object that an accounts payable manager can manage by payment function for any employee expenses in the payment process.

Note

Some data security policies are not defined as grants but directly in applications code. The security reference manuals for Oracle Fusion Applications offerings differentiate between data security policies that define a grant and data security policies defined in Oracle Fusion applications code.

A business object participating in a data security policy is the database resource of the policy.

Data security policies that use job or duty roles refer to data security entitlement.

For example, the data security policy for the Accounts Payable Manager job role refers to the view action on AP disbursements as the data security entitlement.

Important

The duty roles inherited by the job role can be moved and job roles reassembled without having to modify the data security.

As a security guideline, data security policies based on user session context should entitle a duty role. This keeps both function and data security policies at the duty role level, thus reducing errors.

For example, a Sales Party Management Duty can update Sales Party where the provisioned user is a member of the territory associated with the sales account. Or the Sales Party Management Duty can update Sales Party where the provisioned user is in the management chain of a resource who is on the sales account team with edit access. Or the Participant Interaction Management Duty can view an Interaction where the provisioned user is a participant of the Interaction.

For example, the Disbursement Process Management Duty role includes entitlement to build documents payable into payments. The Accounts Payable Manager job role inherits the Disbursement Process Management Duty role. Data security policies for the Disbursement Process Management Duty role authorize access to data associated with business objects such as AP disbursements within a business unit. As a result, the user provisioned with the Accounts Payable Manager job role is authorized to view AP disbursements within their business unit.

A data security policy identifies the entitlement (the actions that can be made on logical business objects or dashboards), the roles that can perform those actions, and the conditions that limit access. Conditions are readable WHERE clauses. The WHERE clause is defined in the data as an instance set and this is then referenced on a grant that also records the table name and required entitlement.

Data Roles

Data roles are implemented as job roles for a defined set of data.
A data role defines a dimension of data within which a job is performed. The data role inherits the job role that describes the job. For example, a data role entitles a user to perform a job in a business unit.

The data role inherits abstract or job roles and is granted data security privileges. Data roles carry the function security privileges inherited from job roles and also the data security privilege granted on database objects and table rows.

For example, an accounts payables specialist in the US Business Unit may be assigned the data role Accounts Payables Specialist - US Business Unit. This data role inherits the job role Accounts Payables Specialist and grants access to transactions in the US Business Unit.

A data role may be granted entitlement over a set people.

For example, a Benefits Administrator A-E is allowed to administer benefits for all people that have a surname that begins with A-E.

Data roles are created using data role templates. You create and maintain data roles in the Authorization Policy Manager (APM). Use the Manage Data Roles and Security Profiles task to create and maintain HCM data roles in Oracle Fusion HCM.

**HCM Security Profiles**

HCM security profiles are used to secure HCM data, such as people and departments. You use HCM security profiles to generate grants for an enterprise role. The resulting data role with its role hierarchy and grants operates in the same way as any other data role.

For example, an HCM security profile identifies all employees in the Finance division.

Oracle Fusion Payroll uses HCM security profiles to secure project organizations. Applications outside of HCM can use the HCM Data Roles UI pages to give their roles access to HR people.

**Masking and Encryption**

Oracle Fusion Applications uses masking to protect sensitive data from view by unauthorized users. Encryption APIs mask sensitive fields in applications user interfaces. Additionally, Oracle Data Masking is available for masking data in non-production instances and Oracle Transparent Data Encryption is available for protecting data in transit or in backups independent of managing encryption keys.

---

**Database Resources and Data Security Policies: How They Work Together**

A data security policy applies a condition and allowable actions to a database resource for a role. When that role is provisioned to a user, the user has access to data defined by the policy. In the case of the predefined security reference implementation, this role is always a duty role. Data roles generated to inherit
the job role based on data role templates limit access to database resources in a particular dimension, such as the US business unit.

The database resource defines and instance of a data object. The data object is a table, view, or flexfield.

The following figure shows the database resource definition as the means by which a data security policy secures a data object. The database resource names the data object. The data security policy grants to a role access to that database resource based on the policy’s action and condition.

Database Resources

A database resource specifies access to a table, view, or flexfield that is secured by a data security policy.

- Name providing a means of identifying the database resource
- Data object to which the database resource points

Data Security Policies

Data security policies consist of actions and conditions for accessing all, some, or a single row of a database resource.

- Condition identifying the instance set of values in the data object
- Action specifying the type of access allowed on the available values

Note

If the data security policy needs to be less restrictive than any available database resource for a data object, define a new data security policy.
**Actions**

Actions correspond to privileges that entitle kinds of access to objects, such as view, edit, or delete. The actions allowed by a data security policy include all or a subset of the actions that exist for the database resource.

**Conditions**

A condition is either a SQL predicate or an XML filter. A condition expresses the values in the data object by a search operator or a relationship in a tree hierarchy. A SQL predicate, unlike an XML filter, is entered in a text field in the data security user interface pages and supports more complex filtering than an XML filter, such as nesting of conditions or sub queries. An XML filter, unlike a SQL predicate, is assembled from choices in the UI pages as an AND statement.

**Tip**

An XML filter can be effective in downstream processes such as business intelligence metrics. A SQL predicate cannot be used in downstream metrics.

---

**Securing Data Access: Points to Consider**

Oracle Fusion Applications supports securing data through role-based access control (RBAC) by the following methods.

<table>
<thead>
<tr>
<th>Method of securing data</th>
<th>Reason</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data roles apply explicit data security policies on job and abstract roles</td>
<td>Appropriate for job and abstract roles that should only access a subset of data, as defined by the data role template that generates the data role or by HCM security profiles.</td>
<td>Accounts Payable Manager - US data role to provide an accounts payable manager in the US business unit with access to invoices in the US business unit.</td>
</tr>
</tbody>
</table>

If a user has access to the same function through different roles that access different data sets, then the user has access to a union of those data sets.

When a runtime session is created, Oracle Platform Security Services (OPSS) propagates only the necessary user to role mapping based on Oracle Fusion Data Security grants. A grant can specify entitlement to the following:

- Specific rows of data (data object) identified by primary key
- Groups of data (instance set) based on a predicate that names a particular parameter
- Data objects or instance sets based on runtime user session variables
Data is either identified by the primary key value of the row in the table where the data is stored. Or data is identified by a rule (SQL predicate) applied to the WHERE clause of a query against the table where the data is stored.

Grants

Oracle Fusion Data Security can be used to restrict the following.

- Rows that are returned by a given query based on the intended business operation
- Actions that are available for a given row

Grants control which data a user can access.

Note

Attribute level security using grants requires a data security policy to secure the attribute and the entitlement check enforces that policy.

A grant logically joins a user or role and an entitlement with a static or parameterized object instance set. For example, \texttt{REGION='WEST'} is a static object instance set and \texttt{REGION=&GRANT_ALIAS.PARAMETER1} is a parameterized object instance set. In the context of a specific object instance, grants specify the allowable actions on the set of accessible object instances. In the database, grants are stored in FND_GRANTS and object instance sets are stored in FND_OBJECT_INSTANCE_SETS. Object access can be tested using the privilege check application programming interface (API).

Securing a Business Object

A business object is a logical entity that is typically implemented as a table or view, and corresponds to a physical database resource. The data security policies of the security reference implementation secure predefined database resources. Use the Manage Data Security Policies task to define and register other database resources.

Data security policies identify sets of data on the registered business object and the actions that may be performed on the business object by a role. The grant can be made by data instance, instance set or at a global level.

Note

Use parameterized object instance sets whenever feasible to reduce the number of predicates the database parses and the number of administrative intervention required as static object instances sets become obsolete. In HCM, security profiles generate the instance sets.

Data Role Templates: Explained

You use data role templates to generate data roles. You generate such data roles, and create and maintain data role templates in the Authorization Policy Manager (APM).

13-28 Oracle Fusion Applications Cost Accounting and Receipt Accounting Implementation Guide
Note

HCM data roles are generated using the Manage Data Roles and Security Profiles task, which uses HCM security profiles, not data role templates, to define the data security condition.

The following attributes define a data role template.

- Template name
- Template description
- Template group ID
- Base roles
- Data dimension
- Data role naming rule
- Data security policies

The data role template specifies which base roles to combine with which dimension values for a set of data security policies. The base roles are the parent job or abstract roles of the data roles.

Note

Abstract, job, and data roles are enterprise roles in Oracle Fusion Applications. Oracle Fusion Middleware products such as Oracle Identity Manager (OIM) and Authorization Policy Manager (APM) refer to enterprise roles as external roles. Duty roles are implemented as application roles in APM and scoped to individual Oracle Fusion Applications.

The dimension expresses stripes of data, such as territorial or geographic information you use to partition enterprise data. For example, business units are a type of dimension, and the values picked up for that dimension by the data role template as it creates data roles are the business units defined for your enterprise. The data role template constrains the generated data roles with grants of entitlement to access specific data resources with particular actions. The data role provides provisioned users with access to a dimensional subset of the data granted by a data security policy.

An example of a dimension is a business unit. An example of a dimension value is a specific business unit defined in your enterprise, such as US. An example of a data security policy is a grant to access a business object such as an invoice with a view entitlement.

When you generate data roles, the template applies the values of the dimension and participant data security policies to the group of base roles.

The template generates the data roles using a naming convention specified by the template’s naming rule. The generated data roles are stored in the Lightweight Directory Access Protocol (LDAP) store. Once a data role is generated, you provision it to users. A user provisioned with a data role is granted permission to access the data defined by the dimension and data security grant policies of the data role template.
For example, a data role template contains an Accounts Payable Specialist role and an Accounts Payable Manager role as its base roles, and region as its dimension, with the dimension values US and UK. The naming convention is [base-role-name]:[DIMENSION-CODE-NAME]. This data role template generates four data roles.

- Accounts Payable Specialist - US (business unit)
- Accounts Payable Specialist - UK (business unit)
- Accounts Payable Manager - US (business unit)
- Accounts Payable Manager - UK (business unit)

Making Changes To Data Role Templates

If you add a base role to an existing data role template, you can generate a new set of data roles. If the naming rule is unchanged, existing data roles are overwritten.

If you remove a base role from a data role template and regenerate data roles, a resulting invalid role list gives you the option to delete or disable the data roles that would be changed by that removal.

Making Changes to Dimension Values

If you add a dimension value to your enterprise that is used by a data role template, you must regenerate roles from that data role template to create a data role for the new dimension. For example if you add a business unit to your enterprise, you must regenerate data roles from the data role templates that include business unit as a dimension.

If you add or remove a dimension value from your enterprise that is used to generate data roles, regenerating the set of data roles adds or removes the data roles for those dimension values. If your enterprise has scheduled regeneration as an Oracle Enterprise Scheduler Services process, the changes are made automatically.

For information on working with data role templates, see the Oracle Fusion Middleware Administrator’s Guide for Authorization Policy Manager (Oracle Fusion Applications Edition).

Define Document Sequences

Document Sequences: Explained

In Oracle Fusion Applications, each business document or business event is uniquely identified by a document sequence number that you assign to it. However, the document sequencing feature must be turned on (enabled) on the business document or event to allow the assignment. For example, if document sequencing is enabled, you can assign a document sequence number.
to an invoice that gets generated in response to a purchase order. You can use
document sequences as a proof to track successfully executed transactions
as well as failed transactions. Additionally, a document sequence helps in
generating an audit trail, which can be used to identify how a particular
transaction passed through various applications.

Document sequencing can be managed automatically, manually, and gaplessly.

---

**Note**

Plan your document sequencing carefully before you use the options available
in the application to apply sequence numbers. Avoid changes to the options
after you saved your work on the Manage Document Sequences and Manage
Document Sequence Categories pages.

---

**Automatic Sequencing**

Automatic document sequencing assigns a unique number to each document
as it is generated, and this unique number is stored in the database. The
numbering is sequential by date and time of creation. If you define a sequence to
automatically number documents, you can provide an initial value to begin the
sequence. In absence of a custom value, the default value 1 is used.

**Manual Sequencing**

Manual sequencing requires you to assign a unique number to each document
before it is generated. In manual sequencing, the numerical ordering and
completeness of a transaction is not enforced. Users can skip or omit numbers
when entering the sequence value. However, each time that a number is
assigned, the application validates its uniqueness.

**Gapless Sequencing**

Gapless sequencing is similar to automatic sequencing. It automatically
generates a unique number for each document, but does that only for
successfully generated documents. As a result, the sequence is maintained for
all the documents that are generated, and no sequence numbers are lost due to
incomplete or failed document generation.

---

**Important**

Use this type of sequencing only if necessary because it may affect the
performance of the system and slow down transaction processing.

---

**Sequential Numbering Enforced Profile Options**

The Sequential Numbering Enforced profile validates the documents being
created and applies the relevant document sequence, based on the selected
option. The following options are available:

- **Always Used**: Sequential numbering is enforced for all document
categories. When this option is set, users cannot create a document if its
document category is not assigned any active document sequence. The
system displays an error.

- **Partially Used**: Sequential numbering is enforced for all document
categories. However, if no sequence is available for the document, the
system displays a warning. Users can either proceed without a document sequence or create and assign a document sequence before proceeding.

- Not Used: Sequential numbering is not enforced. In this mode, although the documents created do not require a document sequence to be assigned, the system checks for the uniqueness of the document number provided. The system does not display any warning or error.

**Restriction**

At the site level, the profile is by default set to Partially Used. Avoid changing this option as doing so may interfere with the validation logic and the data that was stored using the earlier option may not appear. However, you can change the option at the product or user level.

---

**Document Sequence Categories: Explained**

A document sequence category is a set of documents that share similar characteristics and that are formed into a logical group. Document sequence categories simplify the task of assigning number sequences to specific documents. Instead of assigning a number to each document, you assign a document sequence to one or more document sequence categories. The document sequence category automatically takes care of numbering the documents.

A document sequence category identifies the database table that stores documents resulting from transactions that your users enter. When you assign a sequence to a category, the sequence numbers the documents that are stored in a particular table. You must create document sequence categories to be able to manage the task of assigning document sequences.

**Restriction**

Once a document sequence category is created, you cannot change the application, the category code, or the table name. Therefore, carefully consider these details and plan your document sequencing requirement before you begin working with the application.

Once you create a document sequence category, it is available for use under the **Document Sequences: Assignments** section on the Manage Document Sequences page. The **Category** field contains the name of the document sequence category. After you create a document sequence, you can assign it to a document sequence category.

---

**Document Sequences: Points to Consider**

Sequencing documents is an important business and legal requirement. Certain aspects of the defining process are permanent and cannot be modified later. Therefore, it is important that you first decide the appropriate document sequence to use for a set of documents. You must also decide beforehand the type of document sequencing, because you are not allowed to switch to other types once a sequence is assigned to a document sequence category. Make a
note of the details such as the document sequence and document sequence category so that you can refer to them at a later point in time. Also note if there are any restrictions or configuration prerequisites before you define document sequencing.

Note

Products that implement document sequencing have specifications about its usage. Refer to the corresponding product documentation for specific details and also to determine if there are any restrictions or configuration prerequisites.

Creating and Editing Document Sequences

You can create document sequences that are automatic, manual, or gapless, depending on the business or legal requirement. By default, the current date is considered as the start date. If the end date is left blank, it means that the sequence definition never expires. Among the several options used in creating and editing document sequences, the following options are functionally more important and therefore need to be carefully determined:

- **Determinant Type**: Select to limit the document sequencing activity to certain documents that belong to a specific business entity, such as Ledger, Tax Registration, and so on.

- **Initial Value**: Enter a value for the first document in your sequence. This field applies only to sequences with automatic or gapless numbering types. Sequence numbers should not be greater than eight digits. If you leave this field blank, the first document is automatically assigned a value of 1. Once a document sequence is defined, you cannot change this initial value.

Creating and Editing Document Sequence Categories

Document sequence categories are defined to make it easy to assign document sequence definitions to a group of documents instead of to individual documents. Each document sequence category is mapped to a specific table, where the documents belonging to that category are stored. The table must already be enabled for document sequencing. When specifying the table, you must consider the following points:

- When the sequential numbering feature checks for completeness or generates a report, it locates the category’s documents in the table.

- You can select only tables belonging to the application associated with the category.

- Once a category is defined, you cannot change the choice of table.

Assigning Document Sequences

Identify the documents to be numbered before assigning them a document sequence. For each document sequence, there can be only one active assignment to a document sequence category, a method code, and a determinant value (if applicable). As part of the assignment, specify whether the document is created automatically (for example, due to a batch process, or manually through a
form). If you do not specify an end date, the assignment continues to remain active throughout the process cycle. If a determinant type was specified for the document sequence, then enter a specific determinant value related to the selected determinant type.

At runtime, when users create documents, the document sequence to be assigned is determined by finding the active assignment that matches the correct combination of category, numbering method, and the date range containing the transaction date.

**Auditing Document Sequences**

You can audit document sequences, if required, to provide an audit trail of the document sequences used in a specific product. However, before enabling the audit functionality for a document sequence, you must have created an audit table for the specific document sequence, using appropriate details. Enabling the audit functionality is permitted only for newly created document sequences. You cannot audit document sequences that are already in use by a specific product.

For more information about defining a document sequence audit table, see the Oracle Fusion Applications Developer's Guide.

**Define Trees**

**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.

<table>
<thead>
<tr>
<th>Status</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft</td>
<td>Yet to be published or is in a modified state.</td>
</tr>
<tr>
<td>Active</td>
<td>In use and based on which one or more trees or tree versions are created.</td>
</tr>
<tr>
<td>--------</td>
<td>---------------------------------------------------------------------</td>
</tr>
<tr>
<td>Inactive</td>
<td>Not in use.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Validator</th>
<th>Description (what is checked)</th>
<th>Possible Cause for Validation Failure</th>
<th>Suggested Corrective Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restrict By Set ID</td>
<td>On the Manage Tree Structures: Specify Data Sources page, if the Set ID check box is selected to enable the Restrict Tree Node List of Values Based on 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.</td>
<td>Even when the check box is selected, one or more of its data source view objects does not contain a reference data set attribute.</td>
<td>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.</td>
</tr>
<tr>
<td>Row Flattened Table Name</td>
<td>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 FND_TREE_NODE_RF or a custom table.</td>
<td>• The specified table does not exist in the database. • The specified table does not contain the same columns as the FND_TREE_NODE_RF table.</td>
<td>Correct the row flattened table definition.</td>
</tr>
</tbody>
</table>
| Available Label Data Sources | 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 **Labeling Scheme** 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 **None** from the **Labeling Scheme** list box. | • Any of the specified label data source view objects do not exist.  
• Any of the specified label data source view objects do not have primary keys.  
• 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. | • Correct the specified label data source view object.  
• Correct the primary keys of the specified label data source view object.  
• Either correct the primary keys in the label data source view object to match the primary keys that were earlier registered in FND_TS_DATA_SOURCE, or correct the primary keys registered in that table to match the new view object definition. |
| 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. | • Any of the specified data source view objects do not exist.  

• When a data source view object is initially defined, the database automatically registers the primary keys for the view object if the **Use non-defined primary key columns** 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.  

• The **Use non-defined primary key columns** 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.  

• 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. | • Correct the specified data source view object.  

• Correct the duplicate column in the registered primary keys.  

• Correct the primary keys of the specified data source view object.  

• Correct any mismatch in data types. |
<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
<th>Common Issues</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column Flattened Table</td>
<td>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 FND_TREE_NODE_CF or a custom table.</td>
<td>• The specified table does not exist in the database.</td>
<td>Correct the column flattened table definition.</td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td>• The specified table does not contain the same columns as the FND_TREE_NODE_CF table.</td>
<td></td>
</tr>
<tr>
<td>Restrict by Date</td>
<td>On the Manage Tree Structures: Specify Data Sources page, if the <strong>Date Range</strong> check box is selected to enable the <strong>Restrict Tree Node List of Values Based on</strong> 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.</td>
<td>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.</td>
<td>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.</td>
</tr>
<tr>
<td>Tree Node Table Name</td>
<td>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 FND_TREE_NODE or a custom table.</td>
<td>• No table is specified in the <strong>Tree Node Table</strong> field.</td>
<td>Correct the tree node table definition.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The specified table does not exist in the database.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The specified table does not contain the same columns as the FND_TREE_NODE table.</td>
<td></td>
</tr>
<tr>
<td>Allow Node Level</td>
<td>If the Allow Node Level Security 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.</td>
<td>The option is set to No for the tree structure but one or more associated data sources have that option set to Yes.</td>
<td>Correct the option setting in the tree structure and their data sources.</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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 flatteningoptimizes 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.

<table>
<thead>
<tr>
<th>Labeling Scheme</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>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.</td>
</tr>
<tr>
<td>Group</td>
<td>Labels that you can arbitrarily assign to tree nodes.</td>
</tr>
<tr>
<td>Depth</td>
<td>Labels that are automatically assigned based on the depth of the tree node within the tree. No manual assignment is performed.</td>
</tr>
</tbody>
</table>

**Note**
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.

<table>
<thead>
<tr>
<th>Validator</th>
<th>Description (what is checked)</th>
<th>Possible Cause for Validation Failure</th>
<th>Suggested Corrective Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effective Date</td>
<td>The effective start and end dates of the tree version must be valid.</td>
<td>The effective end date is set to a value that is not greater than the effective start date.</td>
<td>Modify the effective start and end dates such that the effective start date is earlier than the effective end date.</td>
</tr>
<tr>
<td>Root Node</td>
<td>On the Manage Tree Structures: Specify Data Sources page, if the Allow Multiple Root Nodes check box for the Restrict Tree Node List of Values Based on 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.</td>
<td>Even if the check box is deselected, the tree version has multiple root nodes.</td>
<td>Modify the tree version such that there is exactly one root node.</td>
</tr>
<tr>
<td>Data Source Max Depth</td>
<td>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 Maximum Depth field is set to Unlimited.</td>
<td>The tree version has data at a depth greater than the specified depth limit on one or more data sources.</td>
<td>Modify the tree version such that all nodes are at a depth that complies with the data source depth limit.</td>
</tr>
<tr>
<td>Duplicate Node</td>
<td>On the Data Source dialog box, if the Allow Duplicates 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.</td>
<td>Even when the check box is deselected, the tree version contains duplicate nodes.</td>
<td>Remove any duplicate nodes from the tree version.</td>
</tr>
</tbody>
</table>
| Available Node | All nodes in the tree version should be valid and available in the underlying data source. | • 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.  
• The tree version contains a tree reference node, which references another tree version that does not exist. | Remove any orphaned nodes from the tree version. Update tree reference nodes so that they reference existing tree versions. |
<p>| 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. |</p>
<table>
<thead>
<tr>
<th>SetID Restricted Node</th>
<th>On the Manage Tree Structures: Specify Data sources page, if the Set ID check box is selected to enable the <strong>Restrict Tree Node List of Values Based on</strong> 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.</th>
<th>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.</th>
<th>Modify the tree version such that all nodes in the tree have data sources with reference data set matching that of the tree.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label Enabled Node</td>
<td>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 <strong>Labeling Scheme</strong> list box, all nodes should have labels. This restriction does not apply when you select <strong>None</strong> from the <strong>Labeling Scheme</strong> list box.</td>
<td>The tree structure has a labeling scheme but the tree version has nodes without labels.</td>
<td>Assign a label to any node that does not have a label.</td>
</tr>
<tr>
<td>Date Restricted Node</td>
<td>On the Manage Tree Structures: Specify Data Sources page, if the <strong>Date Range</strong> check box is selected to enable the <strong>Restrict Tree Node List of Values Based on</strong> 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.</td>
<td>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).</td>
<td>Ensure that all nodes in the tree version have effective date range for the effective date range for the tree version.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Multiple Active Tree Version</td>
<td>On the Manage Tree Structures: Specify Definition page, if the <strong>Allow Multiple Active Tree Versions</strong> 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.</td>
<td>Even when the check box is not selected, there is more than one active tree version in the tree for the same date range.</td>
<td>Set no more than one tree version to Active within the same date range and set the others to inactive or draft status.</td>
</tr>
<tr>
<td>Range Based Node</td>
<td>On the Data Source dialog box, if the <strong>Allow Range Children</strong> 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.</td>
<td>Even when the check box is not selected, there are range-based nodes from a data source.</td>
<td>Ensure that any range nodes in your tree version are from a data source that allows range children.</td>
</tr>
</tbody>
</table>
## Terminal Node

<table>
<thead>
<tr>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>On the Data Source dialog box, if the <strong>Allow Use as Leaves</strong> 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.</td>
<td>Even when the check box is not selected, values from a data source are added as leaf nodes (terminal nodes).</td>
</tr>
<tr>
<td>Modify the tree version such that all terminal nodes are from data sources for which this check box is selected.</td>
<td>For each data source value that is not yet available, add nodes to the tree version.</td>
</tr>
</tbody>
</table>

## Usage Limit

<table>
<thead>
<tr>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>On the Data Source dialog box, if the <strong>Use All Values</strong> option is selected to set the <strong>Usage Limit</strong> for the data source, every value in the data source must appear as a node in the tree. This restriction does not apply if <strong>None</strong> option is selected.</td>
<td>Even if the <strong>Use All Values</strong> option is selected, there are values in the data source that are not in the tree version.</td>
</tr>
</tbody>
</table>

## 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.

**Define Profile Options**

**Profile Options: Explained**

Profile options manage configuration data centrally and influence the behavior of applications. Profile options serve as permanent user preferences and application configuration parameters. You configure profile options with settings for specific contexts or groups of users. Users customize how their user interfaces look and behave by changing the values of available profile options.
Profile options store the following kinds of information.

<table>
<thead>
<tr>
<th>Type of Information</th>
<th>Profile Option Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>User preferences</td>
<td>Settings to provide access to social networking features</td>
</tr>
<tr>
<td>Installation information</td>
<td>Setting to identify the location of a portal</td>
</tr>
<tr>
<td>Configuration choices</td>
<td>Settings to change user interface skins and behaviors</td>
</tr>
<tr>
<td>Processing options</td>
<td>Settings to affect how much information to log either for an entire site or a specific user</td>
</tr>
</tbody>
</table>

You can add and configure new profile options in addition to configuring predefined profile options that are implemented as updatable.

**Profile Option Definition and Configuration**

Application developers add new profile options and configure ones that are not to be updated by other users. Application administrators and implementation consultants configure profile options with profile option values that are implemented as updatable.

Profile option definitions consist of the following.

- Profile option name
- Application and module in the application taxonomy
- Profile option values
- Profile options categories
- Profile option levels
- Profile option level hierarchy

Profile options can appear on any user interface page without indication that a profile option is what is being set.

**Profile Option Values**

Some profile options have predefined profile option values.

The Manage Profile Option Values task flow allows an administrator to set updatable profile option values at the available levels, including the user level. You can access the Manage Profile Option Values task starting in the Setup and Maintenance Overview page and searching for profile option tasks.

You can set profile option values at different levels: site, product, and user. The following table provides examples.

<table>
<thead>
<tr>
<th>Profile Option Level</th>
<th>Value of the Profile Option Level</th>
<th>Profile Option Value</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Manager1</td>
<td>UK pound sterling</td>
<td>Access to site and all products shows UK pounds sterling in effect</td>
</tr>
</tbody>
</table>
Context such as user session or accessed product determines which profile option value is associated with the profile option name. In the example, if manager1 does not set a profile option value for this profile option, access to Financials for EMEA shows currency in Euros; and access to other products shows currency in UK pounds sterling.

**Profile Option Categories**

Categories group profile options based on their functional area. Profile option categories facilitate searching and defining data security.

For example, in Oracle Fusion Receivables, the Transactions profile option category groups profile options related to setting how Receivables transactions are to be processed, such as Require Adjustment Reason.

A profile option can be in more than one category.

**Profile Option Hierarchies and Levels**

Application developers specify at which hierarchy level a profile option is enabled. The predefined profile option hierarchy levels are site, product, and user.

The hierarchy levels specified in the profile option definition determine the context in which a profile option value may be set. If the profile option value at a particular level is updatable, an administrator can update the profile option value for that context.

**Note**

Profile options should only be enabled for context levels that are appropriate for that profile option. For example, a profile option indicating a global configuration setting should not be enabled at the user level, if users cannot choose a different value for that setting.

For security, one level in the hierarchy is designated as a user level. A profile option may be enabled at any or all hierarchy levels. When enabled at all levels, the predefined ordering of profile option hierarchy levels gives precedence to the values that are set at the user level over values set at the product and site levels, and precedence to values set at the product level to values set at the site level. If there is no value for the current user, then the product value applies. If there is no value for the user or product, then the site value applies.

The table shows the predefined profile option hierarchy and ordering.
<table>
<thead>
<tr>
<th>Hierarchy Level</th>
<th>Priority When Multiple Levels Set</th>
<th>Effect on Applications</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site</td>
<td>Lowest</td>
<td>Affect all applications for a given implementation</td>
<td>Currency for the site is set to Euros.</td>
</tr>
<tr>
<td>Product</td>
<td>Supersedes Site</td>
<td>Affect all applications of a product family such as Financials</td>
<td>Currency for the Financials products set to UK pound sterling.</td>
</tr>
<tr>
<td>User</td>
<td>Highest, supersedes Product</td>
<td>Affect only the experience of the current user</td>
<td>Currency for the user of Financials applications set to US dollars.</td>
</tr>
</tbody>
</table>

You can configure updatable values for profile options at one or more levels depending on which levels are enabled in the profile option definition. When a profile is set at more than one level, higher levels of specificity override lower levels of specificity.

In the example, if the currency setting for the site is UK pounds sterling, but the Financials division works in the Netherlands using the Euro, a manager in the US can override that product level setting at the user level to use US dollars when accessing Financials applications.

In another example, if a profile option called Printer is set only at the site and product levels. When a user logs on, the Printer profile option assumes the value set at the product level, since it is the highest level setting for the profile.

**Tip**

Set site-level profile option values before specifying values at any other level. The profile option values specified at the site-level work as defaults until profile option values are specified at the other levels.

---

**Planning Profile Options: Points to Consider**

Plan profile options before defining and configuring them.

The following aspects assist you in better planning how to manage profile options.

- Profile option tasks
- Before creating a profile option
- Profile options data model

**Profile Option Tasks**

Users may be able to set their own profile options, depending on settings in the profile option definition. However, not all profile options are visible to end users, and some profile options, while visible, may not be updated by end users.

The following table lists tasks and considerations relevant to planning profile options.
<table>
<thead>
<tr>
<th>Tasks</th>
<th>Role</th>
<th>Considerations</th>
</tr>
</thead>
</table>
| Planning, creating, and editing a new profile option | Applications developer                                  | Since profile options are for permanent settings, do not use profiles options to cache temporary session attributes.  
Add capacity for user preferences and system configuration. Customize profile options with values, value behaviors, validation, category values, and security. Define the levels at which the profile option is enabled. |
| Configure values in an existing profile option   | Applications developer, application administrator, and implementation consultant | Manage the values for existing profile options.                                                                                                                                                          |
| Create and edit profile option categories       | Applications developer, application administrator, and implementation consultant | Manage categories for organizing existing profile options.                                                                                                                                             |

**Note**

Since a profile option enables a behavior in an application user interface or across applications, a value change made by an end user is reflected in the UI page for managing profile option values.

**Before Creating a Profile Option**

Profile options are best defined for managing configuration data centrally and influencing the behavior of applications.

If the purpose of a profile option setting is specific to a piece of data (typically setup data), it is best implemented as an attribute of that data.

Do not use profile options for behavior that is not configurable.

Profile options exist independent of role.

Do not use profile options to implement function security. For example, an application should not check for a profile option value set to yes to provide access to a page. Do not use profile options to implement data security, such as a profile option value that must be set to a specific value to provide view access to an entity.

Evaluate if there is a genuine need before creating a profile option. Do not force users to make a decision about an aspect of their application use that is of no concern.

Evaluating need includes looking for duplicate or similar profile options, even in other products, before creating a new one. For example, you do not need multiple profile options to choose a preferred currency.
Profile Options Data Model

The profile option data model illustrates the relationships among profile option elements.

The figure shows the data model of profile option entities.

For more information about planning profile options, see the Oracle Fusion Applications Developer’s Guide.

Managing Profile Options: Points to Consider

A profile option definition consists of a name for the profile option and valid values. It is defined within a module of the application taxonomy. Application developers manage profile options to create new profile options or modify existing profile option definitions, which includes specifying the levels at which a profile option is enabled and defining values. Implementation consultants and application administrators configure existing profile options by managing the profile option’s updatable values, and creating categories that group profile options.

Configuring a Profile Option

A profile option definition includes information about the owning application and module in the application taxonomy. A start or end date, or both may limit when a profile option is active. The profile option definition may include an SQL validation statement that determines which values are valid, and the hierarchy levels at which the profile option is enabled and updatable.

To be visible to users, a profile option must be user enabled. You can also allow user updates of the profile option, which means users can make changes to the validation and the profile option level information.
Profile option levels specify at which context level profile values may be enabled or updated.

Profile options should only be enabled for context levels that are appropriate for that profile option. For example, a profile option indicating a global configuration setting should not be enabled at the user level, if users cannot choose a different value for that setting.

**SQL Validation**

The SQL validation of the profile option definition determines what valid profile option values are available. In the absence of validation, any value is valid.

For example, SQL validation provides a means of defining a list of values for the valid values of the profile option. The SQL validation can use lookups to provide the valid values for profile options, such as the lookup codes of the YES_NO lookup type.

With a profile option called DEFAULT_LANGUAGE, you can configure the following validation.

```
SELECT DESCRIPTION Language, NLS_LANGUAGE
FROM FND_LANGUAGES_VL
WHERE INSTALLED_FLAG IN ('B','I')
ORDER BY DESCRIPTION
```

This results in the following list of values based on data in `FND_LANGUAGES_VL`.

<table>
<thead>
<tr>
<th>Display Value</th>
<th>Hidden Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>American English</td>
<td>US</td>
</tr>
<tr>
<td>French</td>
<td>F</td>
</tr>
<tr>
<td>Spanish</td>
<td>E</td>
</tr>
</tbody>
</table>

Hidden values must be varchar2(2000).

Profile options generally provide configuration values within a particular context. Though you can create a profile option to be global, think of global values as default values to avoid storing inappropriate configuration information as profile option values. Create global profile options that have corresponding contextual levels.

**Managing Profile Option Categories: Points to Consider**

Use profile option categories to group profile options.

**Organizing Profile Options in Categories**

As a guideline, group profile options in a single category if the profile options affect the same feature, or if an administrator would likely want to see the profile options in the results of a single search.
Application developers are responsible for the initial groupings and then administrators can make changes based on their specific needs. Administrators can categorize profile options and then easily search on profile options by category.

Tip
Define profile option categories first and assign new profile options to existing categories rather than defining profile options first and then defining categories to categorize them.

Adding New Profile Option Categories

You can add new categories or add profiles to an existing category.

You can create a profile option category by duplicating an existing category and editing it for a new grouping of profile options. You can add multiple profile options to a category. A profile option can exist in multiple categories.

Profile Option Order in a Category

Specify a profile option sequence to determine the order of profile options when queried by profile option category.

Viewing and Editing Profile Option Values: Points to Consider

A profile option value consists of the value and the context or level where the value is set. You specify the context with a pairing of the profile option value’s level and level value, such as the product level and the level value GL for Oracle Fusion General Ledger. Adding or modifying profile option values can include deciding which valid values are enabled or updatable at which level.

The SQL validation of the profile option definition determines what valid profile option values are available. In the absence of validation, any value is valid.

Profile Option Levels and User Session Context

Site level profile option values affect the way all applications run for a given implementation. Product level profile option values affect the way applications owned by a particular product code behave. For example, a product may use profile options set at the product level to determine how regions provided by a common module such as those available from Oracle Fusion Trading Community Model or Customer Relationship Management (CRM) display in a particular work area or dashboard. User level profile option values affect the way applications run for a specific application user

Whichever profile option value is most specific to a user session, that is the value at which the profile option is set for the user session.

For example, the predefined FND_LANGUAGE profile option sets the default language. In addition to a site level value, you can define a value for various product or user levels.
Values at the site level take effect for any user unless overridden by a different value set at the more specific levels of product and user. Product level profile option values affect the way applications owned by a particular product code behave. In addition to user level profile option values in applications, selections may be available in the user preferences workspace.

The following table demonstrates the FND_LANGUAGE profile option settings that would apply to specific users, based on the example above. For example, the user Hima is using the CRM Application Composer product, in the InFusion site. The example above shows that this profile option is set to Hindi at the user level for Hima. Because user is the highest applicable level for Hima, the applicable profile option value is Hindi for Hima.

<table>
<thead>
<tr>
<th>Site</th>
<th>Product</th>
<th>User</th>
<th>Highest Available Level</th>
<th>Active Profile Option Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>InFusion</td>
<td>CRM Application Composer</td>
<td>Hima</td>
<td>User</td>
<td>Hindi</td>
</tr>
<tr>
<td>Acme</td>
<td>Payables</td>
<td>Application Administrator</td>
<td>Product</td>
<td>American English</td>
</tr>
<tr>
<td>InFusion</td>
<td>Payables</td>
<td>Implementation Consultant</td>
<td>Site</td>
<td>American English</td>
</tr>
<tr>
<td>Acme</td>
<td>Payables</td>
<td>Implementation Consultant</td>
<td>none</td>
<td>no value</td>
</tr>
</tbody>
</table>

Note
More than one site level value is relevant in an enterprise with multiple tenants using a single instance of Oracle Fusion Applications.

Effect of Changes to Profile Option Values

Any change you make to a user level profile option has an immediate effect on the way applications run for that session. When you sign in again, changes made to your user level profile options in a previous session are still in effect. When you change profile option value at the product level and no user level values are set, you see the update immediately, but other users may not see the changed value until signing out and back in. When you change a profile option value and the new value affects other users, the change takes effect only when users sign in the next time.

Changes to site level profile options take effect for any user session that is started after the setting has been changed. Changes to site or user level profile options
do not affect any user sessions that are already in progress when the change is made.

Changes to site or user level profile options take effect for any C or PL/SQL processes, such as scheduled jobs, that are launched after the setting has been changed. Profile option changes do not affect C or PL/SQL processes that are already running.

Define Flexfields

Flexfields: Overview

A flexfield is an extensible data field that is divided into segments and available for capturing enterprise specific information. Each segment captures a single atomic value, which is represented in your application database as a single column.

Flexfields allow enterprise requirements to be met without changing the data model. Different data can be captured on the same database table.

The following three types of flexfields are available in Oracle Fusion Applications and provide a means to customize applications features without programming.

- Key
- Descriptive
- Extensible

Flexfields encapsulate all of the pieces of information related to a specific purpose, such as a key identifying a particular purchase, or the components of a student’s contact information, or the features of a product in inventory. Setup of key flexfields is generally required for correct operations of a product. In the case of descriptive and extensible flexfields, segments store attributes on an entity to capture additional information, so setup is usually optional.

End users see flexfield segments as attributes of information displayed in the application user interface. They enter a value for the attribute. End users do not modify the configuration of attributes; they enter values only for attributes that are already configured.

Segments

All flexfields consist of segments.Segments represent attributes of information. They can appear globally wherever the flexfield is implemented, or based on a structure or context.

Value Sets

A value set is a predefined, named group of values that can be used to validate the content of a flexfield segment. The value set you assign to a flexfield segment defines the valid values for the attribute represented by that segment.
Structure

Key flexfields have structure, or a specific configuration of segments. Adding or removing segments, or rearranging their order, produces a different structure. A key flexfield can have multiple structures if registered to support more than one structure. Each instance of a structure shares the same number and order of segments, but differs in the allowable values or value sets that validate the segments.

In some applications, different users need to see different segment structures for the same flexfield. For example, the correctly formatted local postal address for customer service inquiries differs based on locale. The postal address key flexfield can display different segments and prompts for different end users based on a data condition in your application data, such as the user’s role or a value entered by the user.

Context

Descriptive flexfield segments can be context-sensitive and extensible flexfield segments must be context-sensitive. Segments are made available to an application as groups of attributes called contexts. A context is a set of context-sensitive segments that store a particular type of related information.

You define contexts as part of configuring a descriptive or extensible flexfield.

The database columns on which context-sensitive segments are based can be reused in as many contexts as desired.

The same column can be used by a similar segment in different contexts. For example, you can define a Dimensions context that consists of segments representing height, width and depth. You can also define a Measurements context that contains segments which reuse the same underlying height, width and depth columns for length values on an x, y, and z axis, and which additionally includes segments for weight, volume and density.

The same underlying column can be used by different segments in different contexts. For example, you can define a Dimensions context that uses the ATTRIBUTE1 column for height, the ATTRIBUTE2 column for width, and the ATTRIBUTE3 column for depth. You can also define a Measurements context that uses the same columns for other attributes: the ATTRIBUTE1 column for weight, the ATTRIBUTE2 column for volume, and the ATTRIBUTE3 column for density.

Usage

Usage affects various aspects of flexfields. The usage of the flexfield is set when the flexfield is registered and specifies the application and table with which the flexfield is associated.

Entity usage indicates the table containing the segments of a flexfield.

A flexfield can have multiple usages. The first table registered for a flexfield is the master usage. Segments are based on the master usage, and other usages of the same table for the same flexfield use the same segment setup, though the column names optionally may have a differentiating prefix.

You can configure different behavior for extensible flexfield contexts at the usage level. The usage of an extensible flexfield context determines in which scenarios
or user interfaces the segments of a context appear to end users. For example, if a Supplier page displays an extensible flexfield’s supplier usage and a buyer page displays that same extensible flexfield’s buyer usage, a context that is associated to the supplier usage but not the buyer usage displays only on the supplier page and not the buyer page.

The usage of value sets specifies the segments where the value set is assigned.

**Deployment**

A flexfield must be deployed to display its current definition in a runtime application user interface. For example, if the deployment status is Edited, the flexfield segments may already appear in the UI based on the flexfield definition at the time of last deployment.

You can deploy a flexfield as a sandbox for testing the configuration before deploying it to the mainline for test or production users.

**Runtime Appearance**

In an application user interface, flexfield segments appear as label and field pairs or as a table of fields where the column headers correspond to the labels. The fields represent the flexfield segments and accept entered input or a selection from a list of choices that correspond to the segment's assigned value set. Extensible flexfield segments appear grouped within labeled regions, where each grouping is a context and the region labels are the context names.

All segments of a single flexfield are grouped together by default. The layout and positions of the flexfield segments depend on where the application developer places the flexfield on the page. Flexfields may also be presented in a separate section of the page, alone in a table, or on their own page.

You can use Oracle Composer to edit the layout, position, or other display features of the flexfield segments.

Manage flexfields using the tasks of the Define Flexfields activity, which you can access by starting in the Setup and Maintenance Overview page and searching for flexfield tasks.

**Flexfields and Oracle Fusion Application Architecture: How They Work Together**

Administrators configure flexfield segments to capture data that represents the values of attributes. Flexfield segments represent attributes of entities (business objects). Most business objects are enabled for descriptive flexfields. Some business objects are enabled for extensible flexfields.

The figure shows the layers of a flexfield: the business entity table and metadata in the database, business components that are Application Development Framework (ADF) objects or ADFbc objects derived from the metadata and stored in the Metadata Services Repository (MDS), and the user interface where the input fields defined by the flexfield segments are rendered. The flexfield definition consists of all the metadata defined during configuration and stored in the database.
Application developers create a flexfield and register it so that it is available for configuration. Administrators and implementation consultants configure segments and other properties of the available flexfields. This information is stored as additional flexfield metadata in the database. Deploying the flexfield generates ADF business components based on the flexfield metadata in the database.

**Accessing Flexfields For Configuration**

A flexfield affects the Web Services Description Language (WSDL) schemas exposed by ADF services and used by SOA composites. The Web services that expose base entity data also expose flexfield segment data.

Attributes incorporate into SOA infrastructure (BPEL, Rules) and integrate with Oracle Business Intelligence Enterprise Edition and Extended Spread Sheet Database (ESSbase). This ensures that flexfields are preserved across Oracle Fusion Applications updates.

**Deployment**

The metadata for the flexfield is stored in the application database as soon as you save your configuration changes. Deploying the flexfield generates the ADF business components so that the runtime user interface reflects the latest definition of the flexfield in the metadata.

**Runtime**

For a flexfield to reflect the latest flexfield definition at runtime it must be deployed. The user interface accesses a business object and the deployed
Flexfield definition indicates which business object attributes the flexfield captures values for. If you add display customizations for a flexfield using Oracle Composer, these are customizations on the page so that the same flexfield segments can appear differently on various different pages.

Values entered for segments are validated using value sets.

**Flexfield Management**

**Managing Flexfields: Points To Consider**

Managing flexfields involves, registering, planning, and configuring flexfields.

You plan and configure the registered flexfields provided in your applications by applications developers. How you configure flexfield segments determines how the flexfield segments appear to end users. Optionally, you can customize the UI page to change how the flexfield segments appear to end users on that page.

The figure shows the processes involved in making flexfields available to end users. The tasks in the Define Flexfields activity allow administrators to configure and deploy flexfields. If you deploy a flexfield to a sandbox and decide not to make further changes, you select the flexfield in the Manage Flexfields tasks of the Define Flexfields activity and deploy the flexfield in the mainline so it is available to users.
Registering Flexfields

Application development registers flexfields so they are available to administrators and implementation consultants for configuration.

Application development creates the capacity of database tables to support flexfields so an enterprise can capture specific information about an entity. Many flexfields are predefined in Oracle Fusion Applications.

A flexfield must be registered before it can be configured.

For more information on registering flexfields, see Oracle Fusion Applications: Developer’s Guide.

Planning Flexfields

All flexfields consist of segments which represent attributes of an entity. The values an end user inputs for an attribute are stored in a column of the entity table. Application development enables columns of entity tables for use in flexfields during flexfield registration.

A flexfield must be registered before it can be configured. Before configuring new flexfield segments for your enterprise, be sure to plan their implementation carefully.

For more information on planning flexfield configuration, see Oracle Fusion Applications Extensibility Guide.

Configuring Flexfields

Administrators or implementors configure flexfields so they meet the needs of the enterprise. Some flexfields require configuration to make an application operate correctly.

Configuring a flexfield includes the following.

- Defining value sets against which the values entered by end users are validated
- Defining the structure or context of the segments in the flexfield
- Specifying the prompt, length and data type of each flexfield segment
- Specifying valid values for each segment, and the meaning of each value within the application

Tip

As a flexfield guideline, define value sets before configuring the flexfield, because you assign value sets to each segment as you configure a flexfield.

Some descriptive and extensible flexfields provide parameters, which are attributes of the same or related entity objects. You use parameters to set the initial value or derivation value of an attribute from external reference data, such as a column value or a session variable, rather than from user input.
Note
Adding segments to represent additional attributes is considered a custom task. For more information, see the Oracle Fusion Applications Extensibility Guide.

For more information on adding columns to a table, see the Oracle Fusion Applications Developer’s Guide.

For more information on configuring flexfields for custom attributes, see also the Oracle Fusion Applications Extensibility Guide.

Enabling a Flexfield for Business Intelligence

You can enable key flexfields segment instances and descriptive flexfield segments for business intelligence if the flexfield is registered in the database as an Oracle Business Intelligence-enabled flexfield. For more information on enabling segments for business intelligence, see points to consider for managing key and descriptive flexfields.

If you BI-enable multiple segment instances from the same flexfield, the business components of the flexfield are flattened automatically to include a separate attribute for each of the BI enabled segment instances, even if some of them serve the same purpose in your application. You can prevent this duplication and the extra workload and complexity that result, by enabling equalization for any set of segment instances that serve the same purpose in different structure instances. Segment instances that are equalized will appear in the flattened flexfield as a single attribute.

Deploying Flexfields

Once you have configured a flexfield, you must deploy it to make the latest definition available to end users.

You deploy a flexfield in the mainline for general use in a test or production environment, or you can deploy a flexfield as a flexfield-enabled sandbox to confirm that it is correctly configured before deploying it to the mainline.

Deploying a flexfield results in a deployment status. Once deployed, the deployment status indicates the state of the currently configured flexfield relative to the last deployed definition.

Optionally Changing How Flexfield Segments Appear in a User Interface Page

The flexfield attributes you define integrate with the user interface pages where users access the attributes’ business object. Application development determines the UI pages where business objects appear and the display patterns used by default to render flexfield segments.

You can customize the appearance of the flexfield segments in the UI page using Oracle Composer once the flexfield is deployed to the mainline.

For more information on customizing flexfield appearance with Oracle Composer, see guidance on customizing existing pages in Oracle Fusion Applications Extensibility Guide.
For more information on planning flexfields or customizing them beyond the configuration possible in the Define Flexfields tasks, see Oracle Fusion Applications Extensibility Guide.

For more information on creating flexfields and adding them to a UI page, see Oracle Fusion Applications Developer’s Guide.

**Flexfield Segment Properties: Explained**

Independent of the value set assigned to a segment, segments may have properties such as validation.

**Range Validation of Segments**

Range validation enables you to enforce an arithmetic inequality between two segments of a flexfield. For example, a product must be ordered before it can be shipped, so the order date must be on or before the ship date, and consequently the order date segment value must be less than or equal to the ship date segment value. You can use range validation to ensure this relationship.

The conditions for range validation are as follows.

- Segments must be configured for range validation in pairs, one with the low value and one with the high value.
- Both segments must be the same data type.
- Both segments must be part of the same structure in a key flexfield or part of the same context in a descriptive flexfield or extensible flexfield.
- The low value segment must have a lower sequence number than the high value segment.
- Non-range validated segments can exist between a range validated pair, but range validated pairs cannot overlap or be nested.

You can configure as many range validated pairs as you want within the same flexfield. Your application automatically detects and applies range validation to the segment pairs that you define, in sequence order. It must encounter a low value segment first, and the next range validated segment it encounters must be a high value segment. These two segments are assumed to be a matching pair. The low value and the high value can be equal.

**Flexfields and Value Sets: How They Work Together**

Value sets are specific to your enterprise. When gathering information using flexfields, your enterprise’s value sets validate the values your users enter based on how you defined the value set.

You can assign a value set to any number of flexfield segments in the same or different flexfields. Value set usage information indicates which flexfields use the value set.
Defining and Assigning Value Sets

As a flexfield guideline, define value sets before configuring the flexfield, because you assign value sets to each segment as you configure a flexfield.

Caution

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

Shared Value Sets

When you change a value in a shared value set, the change affects the value set for all flexfields that use that value set. The advantage of a shared values set is that a single change propagates to all usages. The drawback is that the change shared across usages may not be appropriate in every case.

Deployment

When you deploy a flexfield, the value sets assigned to the segments of the flexfield provide end users with the valid values for the attributes represented by the segments.

Defaulting and Deriving Segment Values: Explained

To populate a flexfield segment with a default value when a row is created, specify a default type of constant or parameter and a default value.

To synchronize a segment’s value with another field’s value whenever it changes, specify the derivation value to be the flexfield parameter from which to derive the attribute’s value. Whenever the parameter value changes, the attribute’s value is changed to match. If you derive an attribute from a parameter, consider making the attribute read-only, as values entered by users are lost whenever the parameter value changes.

When defaulting or deriving a default value from a parameter, only those attributes designated by development as parameters are available to be chosen.

Different combinations of making the segments read only or editable in combination with the default or derivation value or both, have different effects.

Initial runtime behavior corresponds to the row for the attribute value being created in the entity table. If the default value is read only, it cannot subsequently be changed through the user interface. If the default value is not read only, users can modify it. However, if the segment value is a derived value, a user-modified segment value is overwritten when the derivation value changes.

<table>
<thead>
<tr>
<th>Default Type</th>
<th>Default value specified?</th>
<th>Derivation value specified?</th>
<th>Initial runtime behavior</th>
<th>Runtime behavior after parameter changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>No</td>
<td>Yes</td>
<td>No initial segment value</td>
<td>Changed parameter derivation value updates segment value</td>
</tr>
<tr>
<td>Constant</td>
<td>Yes</td>
<td>No</td>
<td>Default segment value</td>
<td>N/A</td>
</tr>
<tr>
<td>Constant</td>
<td>Yes</td>
<td>Yes</td>
<td>Default segment value</td>
<td>Changed parameter derivation value updates segment value</td>
</tr>
<tr>
<td>--------------</td>
<td>-----</td>
<td>-----</td>
<td>------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>Parameter</td>
<td>Yes</td>
<td>No</td>
<td>Default segment value is the parameter's default value</td>
<td>N/A</td>
</tr>
<tr>
<td>Parameter</td>
<td>Yes</td>
<td>Yes, and same as default value</td>
<td>Default segment value is parameter's default and derivation value</td>
<td>Changed parameter derivation value updates segment value</td>
</tr>
<tr>
<td>Parameter</td>
<td>Yes</td>
<td>Yes, and different from default value</td>
<td>Default segment value is parameter's default value</td>
<td>Changed parameter default value does not update segment value. Only the changed derivation value updates the segment value.</td>
</tr>
</tbody>
</table>

**Flexfield Deployment**

**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.

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.

<table>
<thead>
<tr>
<th>Deployment Status</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edited</td>
<td>The flexfield metadata definition has not been deployed yet. Updates of the metadata definition are not applied in the runtime environment yet.</td>
</tr>
<tr>
<td>Patched</td>
<td>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.</td>
</tr>
<tr>
<td>Deployed to Sandbox</td>
<td>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.</td>
</tr>
<tr>
<td>Deployed</td>
<td>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.</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Error</td>
<td>The deployment attempt in the mainline failed.</td>
</tr>
</tbody>
</table>

**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.

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.

**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.

<table>
<thead>
<tr>
<th>WebLogic Server Tool Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deployFlexForApp</td>
<td>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.</td>
</tr>
<tr>
<td>deployFlex</td>
<td>Deploy a single flexfield regardless of deployment status</td>
</tr>
<tr>
<td>deployPatchedFlex</td>
<td>Deploys flexfield changes that have been delivered using a flexfield Seed Data Framework (SDF) patch. Deploys flexfields that have a Patched deployment status.</td>
</tr>
<tr>
<td>deleteFlexPatchingLabels</td>
<td>Displays MDS label of flexfield changes for viewing and deleting patching labels.</td>
</tr>
</tbody>
</table>

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_shortname` with the application’s short name wrapped in single-quotes.

```
deployFlexForApp('product_application_shortname', ['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:

\[
\text{deployFlexForApp(applicationShortName='product_application_shortname', force='true')}\]

**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 \texttt{flex_code} with the code that identifies the flexfield, and replacing
\texttt{flex_type} with the flexfield's type, which is either DFF, KFF, or EFF.

\[
\text{deployFlex('flex_code', 'flex_type')}\]

The values must be wrapped in single-quotes.

**Using the deployPatchedFlex Command**

Use the \texttt{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.

\[
\text{deployPatchedFlex()}\]

Execute the following command to deploy all flexfields that have either a
READY status or an ERROR status.

\[
\text{deployPatchedFlex(mode='RETRY')}\]

**Using the deleteFlexPatchingLabels Command**

Whenever you deploy flexfield changes to MDS using the
\texttt{deployPatchedFlex()} WLST command, an MDS label is created in the format
FlexPatchingWatermarkdate+time. Use the \texttt{deleteFlexPatchingLabels} command to
inquire about and delete these labels.

From the WLST tool, execute the \texttt{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:

```python
deleteFlexPatchingLabels(infoOnly='true')
```

**Exiting the WLST and Checking the Results**

To exit the tool, execute the following command.

```python
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 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.

Defining Value Sets: Critical Choices

Validation and usage of value sets determine where and how end users access valid values for attributes represented by flexfield segments.

Tip
As a flexfield guideline, define value sets before configuring the flexfield, because you assign value sets to each segment as you configure a flexfield.

Value Sets for Context Segments

When assigning a value set to a context segment, you can only use table-validated or independent value sets. The data type must be character and the
maximum length of the values being stored must not be larger than column length of the context.

Format Only Validation

The format only validation type enables end users to enter any value, as long as it meets your specified formatting rules. That is, the value must not exceed the maximum length you define for your value set, and it must meet any format requirements for that value set.

For example, if the value set allows only numeric characters, your user could enter the value 456 (for a value set with maximum length of three or more), but could not enter the value ABC. A format only value set does not otherwise restrict the range of different values that users can enter. For numeric values, you can also specify if a numeric value should be zero filled or how many digits should follow the radix separator.

Interdependent Value Sets

You cannot specify a dependent value set for a given segment without having first defined an independent value set that you apply to another segment in the same flexfield. You use a dependent value set to limit the list of values for a given segment based on the value that the end user has chosen for a related independent segment. The available values in a dependent list and the meaning of a given value depend on which value was selected for the independently validated segment.

For example, you could define an independent value set of U.S. states with values such as CA, NY, and so on. Then you define a dependent value set of U.S. cities, with values such as San Francisco and Los Angeles that are valid for the independent value CA, and New York City and Albany that are valid for the independent value NY. In the UI, only the valid cities can be selected for a given state.

Because you define a subset value set from an existing independent value set, you must define the independent value set first. End users do not need to choose a value for another segment first to have access to the subset value set.

Table Validation

Typically, you use a table-validated set when the values you want to use are already maintained in an application table (for example, a table of vendor names). Table validation allows you to enable a segment to depend upon multiple prior segments in the same context or structure.

Table-validated value sets have unique values across the table, irrespective of bind variables. The WHERE clause fragment of the value set is considered if it does not have bind variables. If it has bind variables, the assumption is that the values are unique in the value set.

Range

In the case of format, independent, or dependent value sets, you can specify a range to further limit which values are valid. You can specify a range of values.
that are valid within a value set. You can also specify a range validated pair of segments where one segment represents the low end of the range and another segment represents the high end of the range.

For example, you might specify a range for a format-only value set with format type Number where the user can enter only values between 0 and 100. If you use a table value set, you cannot reference flexfield segments in the WHERE clause of the value set. For example, the WHERE clause cannot reference a segment or a value set.

**Security**

In the case of independent and dependent values, you can specify that data security be applied to the values in 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.

When you enable security on a table-validated value sets, the security rule that is defined is absolute and not contingent upon the bind variables (if any) that may be used by the WHERE clause of the value set. For example, suppose a table-validated value set has a bind variable to further filter the value list to x, y and z from a list of x, y, z, xx, yy, zz. The data security rule or filter written against the value set should not assume anything about the bind variables; it must assume the whole list of values is available and write the rule, for example, to allow x, or to allow y and z. By default in data security all values are denied, and show only rows to which access has been provided.

**Maintenance**

There is no need to define or maintain values for a table-validated or subset value set, as the values are managed as part of the referenced table or independent value set, respectively.

If your application has more than one language installed, or there is any possibility that you might install one or more additional languages for your application in the future, select **Translatable**. This does not require you to provide translated values now; but you cannot change this option if you decide to provide them later.

For more information about defining value sets, see the Oracle Fusion Applications Extensibility Guide.

**Manage Descriptive Flexfields**

**Descriptive Flexfields: Explained**

Descriptive flexfields provide a way to add custom attributes to entities, and define validation and display properties for them. These attributes are generally standalone. They don’t necessarily have anything to do with each other and are not treated together as a combination.
All Oracle Fusion Applications business entities that you can access are enabled for descriptive flexfields. Descriptive flexfields are entirely optional. You can choose to configure and expose the descriptive flexfield defined and registered in your database, or not.

A descriptive flexfield provides a set amount of segments for an entity. The segments of a descriptive flexfield are made available to end users as individual fields in the application user interface.

**Segments and Contexts**

Descriptive flexfield segments are of the following types.

<table>
<thead>
<tr>
<th>Segment Type</th>
<th>Runtime Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global segment</td>
<td>Always available</td>
</tr>
<tr>
<td>Context segment</td>
<td>Determines which context-sensitive segments are displayed</td>
</tr>
<tr>
<td>Context-sensitive segment</td>
<td>Displayed depending on the value of the context segment</td>
</tr>
</tbody>
</table>

In the figure, a descriptive flexfield has one context segment called Category for which there are three values: Resistor, Battery, and Capacitor. In addition, the descriptive flexfield consists of two global segments that appear in each of the contexts, and three context-sensitive segments that only appear in the context in which they are configured.

Application development determines the number of segments available for configuring. During implementation you determine which attributes to add using the available segments, and the context values and the combination of attributes in each context. A segment can be used for different attributes, such as Height in Context1 and Color in Context2. Each segment of a descriptive
flexfield that you make available to end users is exposed in the user interface as an individual field.

**Value Sets**

For each global and context-sensitive segment, you configure the values allowed for the segment and how the values that end users enter are validated, including interdependent validation among the segments.

**Managing Descriptive Flexfields: Points To Consider**

Configuring descriptive flexfields involves managing the available flexfields registered with your Oracle Fusion Applications database and configuring their flexfield-level properties, defining and managing descriptive flexfield contexts, and configuring global and context-sensitive segments.

Every descriptive flexfield is registered to include a context segment, which you may choose to use or not.

**Segments**

Assign sequence order numbers to global segments. Segment display is always in a fixed order. You cannot enter a number for one segment that is already in use for a different segment.

Value sets are optional for context segments. The value set that you specify for a context segment consists of a set of context codes, each of which corresponds to a context that is appropriate for the descriptive flexfield. If you do not specify a value set for a context segment, the valid values for that context segment are derived from the context codes. The definition of each context segment specifies the set of context-sensitive segments that can be presented when that context code is selected by the end user.

For reasons of data integrity, you cannot delete an existing context. Instead, you can disable the associated context value in its own value set by setting its end date to a date in the past.

You can configure the individual global segments and context-sensitive segments in a descriptive flexfield. These segment types are differentiated by their usage, but they are configured on application pages that use most of the same properties.

**Usages**

Descriptive flexfield usages allow for the same definition to be applied to multiple entities. Descriptive flexfield tables define the placeholder entity where the flexfield segment values are stored once you have configured the descriptive flexfield.

**Parameters**

Parameters are public arguments to a descriptive flexfield. Parameters provide outside values in descriptive flexfield validation. Parameters can be referenced...
by the logic that derives the default segment value and in table-validated VALUE set WHERE clauses.

**Delimiters**

A segment delimiter or separator visually separates segment values when the flexfield is displayed as a string of concatenated segments.

**Business Intelligence**

If a descriptive flexfield is registered in the database as enabled for Oracle Business Intelligence, you can specify that a global, context, or context-sensitive segment is BI Enabled so it is available for use in Oracle Business Intelligence. The BI Enabled setting is otherwise unavailable.

If you BI-enable a context-sensitive segment, the business components of the flexfield are flattened automatically to include a separate attribute for each of the BI-enabled segments, even if some of them serve the same purpose in your application. If you BI-enable a global segment or context segment, the business components of the flexfield automatically include a single attribute for the global segment or context segment. Descriptive flexfields do not support equalization across context-sensitive segments.

**Manage Extensible Flexfields**

**Extensible Flexfields: Explained**

Extensible flexfields are like descriptive flexfields, with some additional features.

- You can add as many context-sensitive segments to the flexfield as you need. You are not dependent on the number of segments predefined and registered for the flexfield.
- You can configure a one-to-many relationship between the entity and its extended attribute rows.
- A row of data can have multiple contexts associated with it.
- A row of data can have multiple occurrences of the same context.
- You can configure contexts in groups so the attributes in the context always appear together in the user interface.
- You can use existing hierarchical categories so that entities inherit the contexts that are configured for their parents. Contexts are reusable throughout categories.
- You can specify view and edit privileges for the extensible flexfield segments to control who sees the attributes and who can change the attribute's values.

When you configure a context for multiple rows per entity, the segments are displayed as a table.
Unlike descriptive flexfields, the extension columns corresponding to extensible flexfields segments are part of extension tables, separate from the base application table. Unlike descriptive flexfield contexts, the set of attributes in an extensible flexfield context remains constant and does not differ by context value.

An extensible flexfield describes an application entity, with the runtime ability to expand the database that implementation consultants can use to define the data structure that appears in the application.

Extensible flexfields support one-to-many relationships between the entity and the extended attribute rows.

**Usages**

As with descriptive flexfields, you can define multiple usages for an extensible flexfield, which enables several application tables to share the same flexfield.

For example, a flexfield for shipping options can be used by both a Supplier table and a Buyer table. In addition, you can associate a context with one, some, or all of the flexfield’s usages. Thus, with the shipping information example, you can associate a warehouse context with the Supplier usage, a delivery location context with the Buyer usage, and a ship-via context with all usages.

**Categories**

You can define categories for extensible flexfields, and you can associate any combination of contexts with a given category.

For example, the Electronics and Computers category hierarchy might include a Home Entertainment category, which in turn might include an Audio category and a TV category, and so on. The Home Entertainment product might have contexts that specify voltage, dimensions, inputs and outputs. Contexts are reusable within a given extensible flexfield. For example, the dimensions context could be assigned to any category that needs to include dimensional information.

**Pages**

Extensible flexfields allow you to combine contexts into groups known as pages, which serve to connect the contexts so they will always be presented together in the application user interface.

Each application page corresponds to one extensible flexfield category, with a separate region of the page for each associated context.

**Managing Extensible Flexfields: Points To Consider**

Configuring extensible flexfields involves managing the available flexfields registered with your application database and configuring their flexfield-level properties, defining contexts, categories, and pages, and configuring the segments for each extensible flexfield.
Contexts

A context can be defined as single row or multi row. Single row contexts are the same as descriptive flexfields contexts. A single row context has only one set of context-sensitive segments. A multi-row context enables you to associate multiple sets of values with the same object instance.

For example, for a BOOK table, you could create a multi-row context named chapters that contains a chapter segment and a number of pages segment. Multiple chapters can then be associated with each book in the BOOK table.

Set the context to translatable so free-form text entered by end users is stored in the language of the user's locale, and different translations of that text can be stored in other languages. Segments in the translated contexts should utilize format-only value sets for storing free-form, user-entered text.

Categories

A category is a grouping of related data items that can be considered to belong together. You can associate any combination of contexts with a given category.

A category hierarchy logically organizes a set of categories.

For example, the Electronics and Computers category hierarchy might include a Computer category and a Home Entertainment category, which in turn might include an Audio category and a TV category, and so on.

A category can be a child or sibling of an existing category. The hierarchy can be as simple or as complex as desired, with any combination of zero or more sibling categories and zero or more child categories. If no category is defined, the data items are grouped under a single predefined default category.

Each category has associated contexts that store relevant information about a data item in that category. For example, a Home Entertainment product has contexts that specify Voltage, Dimensions, Inputs and Outputs. Contexts are reusable within a given extensible flexfield; the Dimensions context could be assigned to any category that needs to include dimensional information.

If a hierarchy includes child categories, each child category inherits the contexts from its parent category; for example, the Home Entertainment category inherits Voltage and Dimensions from the Electronics and Computers category.

Each extensible flexfield is associated with a particular category hierarchy. You can think of category hierarchies as the defining framework for extensible flexfields and their contexts. A category hierarchy specifies which contexts are valid for each category.

An extensible flexfield can include multiple contexts which you define to support a given category. These contexts can be suitable for a variety of different purposes, but within a particular category, some contexts might be considered to be related to, or dependent on, each other. You can combine these contexts into groups known as pages, which serve to connect the contexts so they will always be presented together in the application user interface.
For example, the Home Entertainment category might have an Electrical Specifications page that contains the Voltage, Inputs and Outputs contexts, and a Physical Specifications page that contains the Dimensions and Form Factor contexts.

Indexed Segments

You can designate an extensible flexfield segment as indexed so that it is one of the selectively required attributes an end user can use in an attribute search. If you indicate in the Manage Extensible Flexfield UI page that a segment should be indexed, the column representing the segment must be added to the database index. This is commonly done by a database administrator (DBA).

When an extensible flexfield with indexed segments is deployed, search task flows are generated along with the other flexfield artifacts and specify the indexed attributes as selectively required. In the deployed extensible flexfield’s search task flow, an end user must specify at least one of the indexed attributes in the search criteria. This prevents non-selective searches which could cause performance issues.

For example, if you index the memory and processor attributes and ensure that the corresponding columns in the database are indexed, an end user can search an item catalog for computers by entering processor or memory or both as a search criterion. No search is performed if an end user enters an attribute as search criteria that is not indexed.

Manage Key Flexfields

Key Flexfields: Explained

Key flexfields provide a means to capture a key such as a part number, a job code, or an account code. A key flexfield consists of one or more segments, where each segment can have a meaning.

For example, a part number 10-PEN-BLA-450 might correspond to a black pen from vendor #450 sold by division #10 (office supplies). Behind the scenes, the application uses a unique number, 13452, for this part, but the end user always see the 10-PEN-BLA-450 part number.

The following aspects are important to understanding key flexfields.

- Architecture
- Segments and segment labels
- Structures
- Segment and structure instances
- Combinations
- Dynamic combination creation
- Security
Key flexfields are not optional. You must configure key flexfields to ensure that your applications operate correctly. You configure and maintain key flexfield definitions with the Manage Key Flexfields task.

Architecture

When you configure a key flexfield, you define metadata about the key flexfield such as how many segments are in a structure, how many structures the flexfield uses, what value sets each segment uses, and so on. This is flexfield metadata stored in flexfield metadata tables.

Based on the flexfield metadata, actual part numbers are captured at runtime as a combination of segment values and stored in a combinations table. A combinations table contains all the segment columns for a flexfield, plus a unique ID column and a structure instance number column that differentiates multiple arrangements of the segment columns.

For example, a part number that can be comprised of multiple segments can be represented by a key flexfield. A part number key flexfield has a corresponding combinations table, where the flexfield stores a list of the complete codes, with one column for each segment of the code, together with the corresponding unique ID and structure instance number for the code. When users define a new part number or maintain existing part numbers in the parts catalog, they directly maintain rows in the combination table.

The foreign key table contains a different business entity than the combinations table. For example, the business entity in the foreign key table is order lines or invoice lines that contain foreign key references to parts for ordering and so on. Any number of foreign key tables can reference a particular entity represented by a key flexfield.

Segments and Segment Labels

A key flexfield consists of segments. Segments consist of a prompt, a short prompt, display width, a number that determines where in the sequence of a key flexfield structure the segment exists, the range type and the column name of the attribute being captured by the segment, a default value set and a label for the segment. A segment label identifies a particular segment of a key flexfield. Segment labels are defined and made available by applications development.

Applications identify a particular segment for some purpose such as security or computations. Segment name or segment order cannot reliably identify a segment because key flexfield segments can be configured to appear in any order with any prompts. A segment label functions as a tag for a segment.

For example, Oracle Fusion General Ledger needs to identify which segment in the Accounting Flexfield contains balancing information and which segment contains natural account information. General Ledger uses a segment label to determine which segment you are using for natural account information. When you define your Accounting Flexfield, you must specify which segment label apply to which segments.

Some labels must be unique, and cannot be applied to more than one segment in each structure. Other labels are required, and must be applied to at least one segment in each structure.
A segment label orients an end user’s search of segments, such as the Cost Center label for all segments across key flexfields that capture a value for cost center.

## Structures

A key flexfield structure definition includes the number of segments and their order.

A delimiter separates the segments when they appear to end users. The delimiter value of a structure specifies the character used to visually separate segment values when the key flexfield is displayed as a string of concatenated segments in the UI.

### Tip

Choose the delimiter value of your key flexfield carefully so that it does not conflict with the flexfield data. For example, if your data frequently contains periods, such as in monetary or numeric values, do not use a period as your segment separator. Any character you expect to appear frequently in your segment values or descriptions is not a good choice for the delimiter.

If you change the configuration of a key flexfield, such as the delimiter, the change affects the previously stored key flexfields with that structure.

Each structure can have one or more segments. Thus a segment is a child of a structure. If you want to store a particular segment, such as Cost Center, in two different structures, you must define the segment separately in each structures.

## Structure and Segment Instances

You can define multiple configurations of a key flexfield structure. These structure instances have the same segment structure, in the same sequence order. They differ primarily in how each segment is validated. You define a structure instance for each key flexfield and each key flexfield structure instance.

The segments in a key flexfield structure instance are segment instances. A segment instance is a segment with a specific value set assigned to it.

If a key flexfield has been registered with a tree structure, you can specify a tree code for a segment instance, where the tree code defines a hierarchical relationship between the segment values.

## Combinations

A combination is a complete code, or combination of segment values that makes up the code, that uniquely identifies an object.

For example, each part number is a single combination, such as PAD-YEL-11x14 or 01-COM-876-7BG-LTN. In these combinations, the hyphen is the segment separator. If you had ten parts you would define ten combinations. A valid combination is simply an existing or new combination that can currently be used because it is not out of date or disabled, and does not violate cross-validation or
security rules. A combination has different segments depending on the flexfield structure being used for that combination. Any combination is associated with only one particular flexfield structure.

Many Oracle Fusion Applications products refer to a key flexfield combination by using the name of the entity or the key flexfield itself. For example, Oracle Fusion Assets uses the asset key flexfield and refers to one of its combinations as an asset key or asset key flexfield. In another example, other Oracle Fusion Applications products including Oracle Fusion General Ledger (GL) refer to combinations of the accounting flexfield as account or GL account.

Each key flexfield has one corresponding table, known as the combinations table, where the flexfield stores a list of the complete codes, with one column for each segment of the code, together with the corresponding unique ID number (a code combination ID number or CCID) for that code. Then, other tables in the application have a column that stores just the unique ID for the code. For example, you may have a part number code, such as PAD-YEL-11x14. The Parts combinations table stores that code along with its ID, 57494. If your application allows you to take orders for parts, you might then have an Orders table that stores orders for parts. That Orders table would contain a single column that contains the part ID, 57494, instead of several columns for the complete code PAD-YEL-11x14.

Typically one combinations page maintains the key flexfield, where the key flexfield is the representation of an entity in your application. The combinations page is where you maintain individual combinations, such as part numbers.

**Dynamic Combination Creation**

Dynamic combination creation is the insertion of a new valid combination into a combinations table from a page other than the combinations page.

Dynamic combination creation may be enabled at the following levels.

<table>
<thead>
<tr>
<th>Level Of Dynamic Combination Creation</th>
<th>Controlled By:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexfield</td>
<td>Application development</td>
</tr>
<tr>
<td>Each usage or reference to the key flexfield</td>
<td>Application development</td>
</tr>
<tr>
<td>Structure instance</td>
<td>Administrators and implementation consultants</td>
</tr>
<tr>
<td>Other</td>
<td>Administrators and implementation consultants</td>
</tr>
</tbody>
</table>

If your key flexfield or certain usages or references of the key flexfield do not permit dynamic combination creation, you may control whether dynamic combination creation is enabled for each structure instance. If enabled, a user can enter a new combination of segment values using the flexfield window from a foreign key page. For example, when entering a transaction, a GL user can enter a new expense account code combination for an account that does not yet exist. Your application creates the new account by inserting the new combination into the combinations table behind the scenes. Assuming that the new combination satisfies any existing cross-validation rules, the flexfield inserts the new combination into the combinations table, even though the combinations table is not the underlying table for the foreign key page.
Managing Key Flexfields: Points to Consider

Consider the plans for a key flexfield, security, and resulting runtime pages when configuring key flexfields.

Planning

Plan structures carefully and allow for future needs.

Caution

Do not change the number, order, and maximum length of segments once you have acquired flexfield data.

Business Intelligence

If a key flexfield is registered in the database as enabled for Oracle Business Intelligence integration, you can specify that a segment instance is BI Enabled, which makes the segment instance available for use in Oracle Business Intelligence. The BI Enabled setting is otherwise unavailable.

If you BI-enable multiple segment instances from the same key flexfield, the business components of the flexfield are flattened automatically to include a separate attribute for each of the BI enabled segment instances, even if some of them serve the same purpose in your application. You can prevent this duplication and the extra workload and complexity that result, by enabling equalization for any set of segment instances that serve the same purpose in different structure instances. Enable equalization by assigning a unique segment label to those segment instances that you wish to be equalized across structure instances. Equalized segment instances appear in the flattened flexfield as a single attribute. Segment instances that are assigned non-unique segment labels are not equalized.

Security

Oracle Fusion data security enforces value set security.

Within key flexfields, value set security applies to the selection of the individual segment values in the segment list of values. When selecting a key flexfield segment value from the combination table, data security allows display of only the combinations whose segment values you have access to. Applications development controls whether or not value set security rules propagate to the foreign key table. By default they do.

Runtime Pages

Application development determines the user interface (UI) pages used to render flexfields. The types of key flexfield UI pages are as follows.

- Combinations pages where underlying entity objects use the combinations table itself
- Foreign key pages where the underlying entity objects contain a foreign key reference to the combinations table
• Partial usage page where some or all of the key flexfield’s segment columns are in a product table

The same key flexfield can be used in different ways on different pages.

A page with a foreign key reference has a base table or view that contains a foreign key reference to a combinations table with the actual flexfield segment columns. This allows manipulating rows containing code combination IDs (CCID).

A page with partial usage of a key flexfield presents segments that are defined on a product’s transactional table in addition to being defined on a combinations table. In the case of a partial usage page, it is possible that only part of the configuration is visible. This allows the key flexfield to behave more like a descriptive flexfield.

A code combination maintenance page or combinations page presents the combinations table. This allows directly creating and maintaining code combinations. The combinations table contains all key flexfield segment columns and a unique ID column.

A typical application has one and only one combinations page. An application might not have a combinations page if it does not support maintenance by administrators.

A page containing a search region enables end users to select which attributes of the key flexfield view object to use as criteria to search for flexfield metadata.

For example, you can configure seven segments for the Account key flexfield. In a foreign key reference page, end users see the typical key flexfield picker with all seven segments where they can search for combinations. In a partial usage page using the same key flexfield, end users potentially could see only a single segment such as the Cost Center labeled segment, or they might see multiple segments but displayed as individual segments rather than as a picker for choosing combinations.

For more information on key flexfield pages, see the Oracle Fusion Applications Developer’s Guide.

**Key Flexfield Structures: Explained**

A key flexfield structure arranges the segments of a key so you can reuse a single key flexfield in multiple combinations of the same or a subset of segments. Multiple instances of a single structure can accommodate differences in the value sets assigned to the structure’s segments.

The structure determines the following aspects of a key flexfield:

• The segments to include

• The order of the segments

• Segment labels on the included segments
• Properties for each segment applied to the instances of the segments in an
instance of the structure

Managing Key Flexfield Structures

All the segments defined for a key flexfield are available to be included in a key
flexfield structure.

You can define as many segments as there are defined segment columns in your
key flexfield combinations table.

Restriction

Be sure to add segments in the order that your key requires. Once deployed, the
order cannot be changed.

Tip

Enable segments to indicate that they are in use. A flexfield does not display
disabled segments in runtime.

Key Flexfield Structure Instances and Segment Instances: Explained

A key flexfield structure can have one or more alternate structure instances.

The instances of a key flexfield structure share the following aspects of the
structure.

• The same set of segments
• The same arrangement of segments
• The same properties at the segment and structure levels

Differences among structure instances at the structure level include whether
dynamic combination creation is allowed.

Differences among segment instances at the structure instance level include the
following.

• Value set
• Default type and default value
• Tree code
• Whether the segment is any of the following
  • Required
- Displayed
- Enabled for business intelligence
- Optional or required as a query criterion

For example, you could use one group of value sets for the US and another for France.

The figure shows two structures instances for a part number structure. The structures differ in the number of segments and the segment separators used. The structure instances of a structure share all properties that are defined for the structure, but can vary in the properties defined at the structure instance or segment instance level, such as the value set assigned to the segment instances.

Query Required Segment Instances

You can designate a key flexfield segment instance as query required so that it is one of the selectively required attributes an end user can use in a key flexfield combination search. If you indicate in the Manage Key Flexfields UI page that a segment instance should be indexed, the column representing the segment must be added to the database index. This is commonly done by a database administrator (DBA).

Following deployment, the combination picker of the key flexfield displays the query required attributes as selectively required. An end user must specify at
least one of the query required attributes in the search criteria. This prevents non-selective searches that could cause performance issues.

For example, if you mark the cost center and account attributes as query required and ensure that the corresponding columns in the database are indexed, an end user can search for combinations by entering cost center or account or both as a search criterion. No search is performed if an end user does not enter at least one query required attribute as search criteria.

**Tip**

Index the Structure Instance Number column on your combinations table to improve performance.

**Dynamic Combinations**

If a key flexfield supports dynamic combination creation, you can choose to enable this feature by selecting **Dynamic Combination Creation Allowed**. This will allow end users to enter values at runtime that produce new code combinations for the flexfield. If not enabled, new valid combinations can only be entered using the combinations table for the flexfield.

**Trees**

If a tree code has been defined for the value set assigned to the segment instance, and you assign the tree code to the segment instance, tree hierarchy search operations are available on the segment values.

For a segment instance to be based on a tree, the following must be true.

- Application development registered the key flexfield with a tree structure.
- A tree code for that tree structure exists.
- The tree code that includes tree versions containing the values of the value set assigned to the segment instance.
- You assign the desired tree code directly to the segment instance.

Provided these conditions are satisfied, different segment instances that use the same value set can be assigned the same or different tree codes, meaning they use a different hierarchy definition over the same values.

**Key Flexfields: Example**

A key flexfield can capture expense account information.

**Scenario**

When entering details for each expense, the user specifies an account to which the expense is charged.
Entering Expense Accounts

A user interface for entering expenses gives the user the option of selecting an expense account that identifies the cost center and other details needed for processing the expense.

Analysis

The expense account field is a foreign key reference to a code combination (EXPENSE_LINES.EXPENSE_ACCOUNT = ACCOUNTS.CCID).

Code Combination Table for Entering Accounts and Employees

The code combination table supports entering account information, such as for expense accounts.

The figure shows the origin in the code combination table of the account specified by the user. The code combination ID record stores the information of the key flexfield segments used to assemble the expense account based on the key flexfield configuration.

The combinations page, which is the maintenance page for the key flexfield, is for managing rows in the combination table. In this example, managing the combinations means adding or editing account numbers that adhere to the key flexfield metadata rules.
The figure shows the code combination details for the example expense account reflected in the flexfield configuration and the code combination table.

If dynamic combination creation is not enabled, then when entering an expense line, the user can only select an account that already exists in the ACCOUNTS (combinations) table. If they require an account that does not exist, they must consult with the appropriate application administrator who can add the account to the combinations table.

If dynamic combination creation is enabled, then when entering an expense line, the user can either select a pre-existing account, or type in a new account that created dynamically on the fly in the ACCOUNTS (combinations) table. Once the new combination is created, the same user can refer to it on the expense line.

When managing employee information, the user specifies the cost center that the employee belongs to. The cost center field corresponds to a single, labeled segment of the Account Key Flexfield and has metadata defined such as the allowable value set for that segment.

In this figure, instead of specifying a cost center ID reference to an account, only the Cost Center segment is used and the value is stored directly on the employee table.
FAQs for Define Flexfields

Why did my flexfield changes not appear in the runtime UI?

The ADF business components or artifacts of a flexfield, which are generated into an metadata services (MDS) repository when the flexfield is deployed, are cached within a user session. You must sign out and back in again to view flexfield definition changes reflected in the runtime application user interface page.

A flexfield’s status relative to its deployment determines whether the flexfield segments as currently defined in the metadata are available to end users. The flexfield segments seen by end users in the runtime correspond to the flexfield definition that was last deployed successfully.

What happens if a value set is security enabled?

Value set security is a feature that enables you to secure access to value set values based on the end user’s role in the system.

As an example, suppose you have a value set of US state names. When this value set is used to validate a flexfield segment, and users can select a value for the segment, you can use value set security to restrict them to selecting only a certain state or subset of states based on their assigned roles in the system.

For example, Western-region employees may choose only California, Nevada, Oregon, and so on as valid values. They cannot select non-Western-region states. Eastern-region employees may choose only New York, New Jersey, Virginia, and so on as valid values, but cannot select non-Eastern-region states. Value set security is implemented using Oracle Fusion Applications data security.

How can I set a default value for a flexfield segment?

When you define or edit a flexfield segment, you specify a default value from the values provided by the value set assigned to that segment.

You can set the default value to be a parameter, which means the entity object attribute to which the parameter you choose is mapped will provide the initial default value for the segment.

You can set to be a constant, if appropriate to the data type of the value set assigned to the segment.

In addition to an initial default value, you can set a derivation value for updating the attribute’s value every time the parameter value changes. The parameter you choose identifies the entity object source attribute. Any changes in the value of the source attribute during runtime are reflected in the value of the segment.

If the display type of the segment is a check box, you can set whether the default value of the segment is checked or unchecked.
Attachments: Explained

Attachments are pieces of supplementary information that users can associate with specific business objects such as expense reports or purchase orders. Attachments can be URLs, desktop files, text, or in cases where available, repository folders. For any given business object, a user may be able to only view attachments, or also create, delete, or edit attachments, depending on security. For more information on an introduction to attachments, see the Oracle Fusion Applications Developer's Guide.

Repository

Attachments are stored in a content server, the Oracle Enterprise Content Management document repository. Users have no real interaction with the repository unless the repository mode is enabled for attachments on specific business objects. In which case, users can share attachments among objects, update attachments by checking them out of and back in to the repository, and perform other tasks. Access to attachment files is controlled by a digital signing mechanism. Depending on security, users might have direct access to the repository.

Security

Data security that applies to a specific business object also applies to attachments for that object, as determined by the attachment entity defined for the object. For example, if a user has no access to a specific expense report, then the same user cannot access attachments for the expense report. You can also use attachment categories to control access and actions on attachments, based on roles associated with the category. For more information on securing attachments, see the Oracle Fusion Applications Developer's Guide.

Attachment Entities: Explained

An attachment entity is usually a database entity, for example a table or view, that represents a business object attachments can be associated with. Each attachment UI must be defined with a corresponding attachment entity, which not only identifies the business object to attach to, but also controls what users can do. Attachment entities are used only in the context of attachments and exist separately from the database entities that they are based on.

Edit and create attachment entities on the Manage Attachment Entities page, which you can access by starting in the Setup and Maintenance Overview page and searching for the Manage Attachment Entities task. Though you would generally use predefined attachment entities with attachment UIs, you might need to create new entities, for example when developing custom UIs.

Entity Names

An attachment entity name should match the name of the table or view that represents the business object to attach to. The name is also used in the
repository folder that is automatically created to store attachments for the entity. The attachment entity display name should be something that users know to represent the business object.

**Database Resource**

The data security policies associated with the database resource defined for the attachment entity would apply to attachments for that entity. For example, based on the database resource for the expense reports attachment entity, the same policies apply to attachments for expense reports. The database resource value must match the value in the OBJ_NAME column in the FND_OBJECTS table for the business object that the entity represents.

**Enabling Security**

Security based on the database resource associated with the attachment entity is always in effect. What you can enable or disable is security based on attachment categories. If any of the attachment categories associated with the attachment entity has data security defined, then that security applies to this entity only if enabled.

**Attachment Entities and Attachment Categories: How They Work Together**

The association between attachment entities and categories determines which categories can be used for an entity. For example, categories associated with the expense report attachment entity are available to be implemented in attachment UIs for expense reports. You can define these associations when managing either entities or categories. Any association changes in either the Manage Attachment Entities or Manage Attachment Categories page are reflected on the other page. You can access either page by starting in the Setup and Maintenance Overview page and searching for attachment tasks.

**Managing Entities**

You determine which attachment categories are relevant to a particular entity on the Manage Attachment Entities page, and each entity must have at least one category. Depending on configuration, any or all of the available categories for that entity are used. For example, you assign three categories to the expense reports attachment entity. For a particular expense report page with attachments functionality, you can customize the attachments component to specify which of the three categories are used. Based on your selection, the data security defined for each category, if any, is applied to attachments on that page if the attachment entity has category-based security enabled.

**Managing Categories**

If you create an attachment category and need to assign it to multiple attachment entities, use the Manage Attachment Categories page. The association means the same as the association on the Manage Attachment Entities page.
Attachments Troubleshooting: Explained

Attachments UIs for users to add and manage attachments are fully functional as is, and users usually would not encounter issues. If you customize attachments in any way, for example by creating additional attachment categories and implementing data security on them, then some issues might arise.

Issue: Unable to View, Add, Update, or Delete Attachments

Users encounter issues when trying to view attachments or perform actions such as adding attachments.

- Users can no longer see specific attachments that they were previously able to see.
- Likewise, they can no longer update or delete attachments.
- Users get an error stating that they do not have permission to add attachments.

Resolution

Use the Manage Attachment Entities page to ensure that attachment categories are associated to the relevant attachment entity. For example, if users can no longer see attachments for an expense report, then search for the expense report attachment entity and assign all necessary categories to it. You might need to check with your system administrator or help desk to determine the exact entity used on the page with the expenses attachments or what categories to assign.

If data security is implemented on the categories for the attachment entity, then verify that the Enable Security check box is selected in the Manage Attachment Entities page for that entity. Make sure that users have a role with the privileges shown in the following table, to view, add, update, or delete attachments with a specific attachment category.

<table>
<thead>
<tr>
<th>Action</th>
<th>Privilege</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>Read Application Attachment</td>
</tr>
<tr>
<td></td>
<td>(FND_READ_APPLICATION_ATTACHMENT_DATA)</td>
</tr>
<tr>
<td>Add or Update</td>
<td>Update Application Attachment</td>
</tr>
<tr>
<td></td>
<td>(FND_UPDATE_APPLICATION_ATTACHMENT_DATA)</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete Application Attachment</td>
</tr>
<tr>
<td></td>
<td>(FND_DELETE_APPLICATION_ATTACHMENT_DATA)</td>
</tr>
</tbody>
</table>

For example, if users have the Read Application Attachment privilege for all categories associated with the expense report attachment entity, except the Receipts attachment category, then they can view all expense report attachments except those created with the Receipts category. Likewise, if users do not have the Update Application Attachment privilege for any attachment categories tied to the expense report attachment entity, then they cannot create any attachments at all for expense reports.
For more information on attachment category data security, see the Oracle Fusion Applications Developer’s Guide.

Finally, certain attachments UI for users have predefined restrictions on categories in place. Your developers can also introduce additional filters to determine which document categories are available for a specific page. Check with your developers or help desk.

**Issue: Missing Attachment Category**

Users can see existing attachments, but the attachments no longer have an attachment category value.

**Resolution**

When the attachment was added, at least one category existed for the corresponding attachment entity, as otherwise the attachment could not have been added. Since then, the entity was edited so that it no longer has any assigned categories, so the user cannot see the category associated with that attachment.

Use the Manage Attachment Entities page to reassign attachment categories to the relevant attachment entity. For example, if users can no longer see the Receipts attachment category for an attachment to an expense report, then search for the expense report attachment entity and assign to it the Receipts category. You might need to check with your system administrator or help desk to determine the exact entity used on the page with the expenses attachments or what additional categories to assign.

Finally, certain attachments UI for users have predefined restrictions on categories in place. Your developers can also introduce additional filters to determine which document categories are available for a specific page. Check with your developers or help desk.

**FAQs for Define Attachments**

**What's an attachment category?**

An attachment category is used to classify and secure attachments. Each attachment user interface must be defined with at least one category for users to be able to add attachments. If there are multiple categories, users can view them and select one when adding attachments. For example, attachments for an expense report can be categorized as receipts, scanned invoice images, and so on.

You can also associate roles with categories to determine user access and actions for attachments, based on the categories assigned to the attachment entity. For example, security for expense report attachments can be based in part on the categories assigned to the expense report attachment entity. You can define multiple categories per module, and add and manage custom categories for your own purposes. For more information on attachment category data security, see the Oracle Fusion Applications Developer’s Guide.

Use the Manage Attachment Categories page, which you can access by starting in the Setup and Maintenance Overview page and searching for the Manage Attachment Categories task.
Set Activity Stream Options

Activity Stream Options: Highlights

Activity Stream is a region on the Oracle Fusion Applications Welcome dashboard and other pages in various applications. Users track the activities and transactions of other users in this region. You can set options that affect the all Activity Stream regions for all users across your site. Individual users can still override your settings through Activity Stream preferences.

Activity stream settings are described in the Oracle Fusion Middleware User's Guide for Oracle WebCenter Spaces. When you read content from that guide, note that:

- Your setup applies to all users, not just yourself or any individual user.
- You can disregard discussions about how to access the settings, because you access the Set Activity Stream Options page by starting in the Setup and Maintenance Overview page and searching for the Set Activity Stream Options task.

Setting Activity Stream Options

- Define the types of users to display activities about in the Activity Stream region, the types of activities to track, and other settings.

See: Setting Activity Stream Preferences

Manage Menu Customizations

Managing Menu Customizations: Highlights

You can customize the Navigator menu, which is the main menu of Oracle Fusion Applications and is always available in the global area. Use the Manage Menu Customizations page, which you can access by starting in the Setup and Maintenance Overview page and searching for the Manage Menu Customization task.

An overview of customizing the Navigator menu is provided in the Oracle Fusion Applications Extensibility Guide. Details of the values you enter to define menu items are provided in the Oracle Fusion Applications Developer's Guide.

Overview

- Navigator menu customization involves managing items, which are nodes in the menu that take the user to the desired destination, and groups, which are categories of items. The label is the menu item or group text displayed to users, and only rendered items and groups are visible to users. Items and groups that are not rendered are displayed in italics in
the menu customization pages. Refer to the Oracle Fusion Applications Extensibility Guide.

See: Customizing the Navigator Menu

**Menu Items**

When you customize items in the Navigator menu, you determine if the item takes users to a specific page in Oracle Fusion Applications or to another application or Web site. The details of what you enter to define the menu item depend on the type of destination. If you duplicate a menu item, the new item appears below the selected source item.

- For menu items to an Oracle Fusion Applications page, do not enter anything in the Destination field, but specify the focus view ID of the target page, Web application, secured resource name, application stripe, and page parameters list. The secured resource name identifies the page definition file that is used to secure resource grants for the page, for example oracle.apps.view.pageDefs.CaseList_Form_Attach_UIShellPagePageDef. The application stripe is the partition for the application in the policy store, for example crm. If you enter both a secured resource name and an application stripe, then the menu item is rendered only if the logged-in user has permission to view that secured resource. If either value is missing, then the item is not secured. Refer to the Oracle Fusion Applications Developer's Guide.

  See: Menu Attributes Added by Oracle Fusion Middleware Extensions for Applications

- For menu items to another application or Web site, enter only the full URL in the Destination field.

**Manage Applications Core Common Reference Objects: Manage Applications Core Messages**

**Common Messages: Points to Consider**

Common messages, which have message names that begin with FND_CMN and message numbers between 0 and 999, are used throughout Oracle Fusion Applications. Each common message can appear in multiple places in any product family. For example, the FND_CMN_NEW_SRCH message can be used for any search to indicate that no results were found. Common messages that are of type error or warning are part of the message dictionary.

**Editing Common Messages**

Because a common message can be used in any application, consider the ramifications if you edit any aspect of the message, including incident and logging settings. Changes would be reflected in all instances where the message is used. For example, if you change the message text, make sure that the text would make sense to all users across Oracle Fusion Applications who might see it.
Creating Common Messages

You can create custom common messages for use in multiple places within a single product. Do not begin the message name with FND_CMN, but use another suitable convention. The message number should be within the range that is designated for the product.

Manage Applications Core Common Reference Objects: Manage Applications Core Administrator Profile Values

Creating and Editing Messages: Highlights

Each message in the message dictionary has many attributes and components, including message properties, text, and tokens, that you define when creating or editing the message.

Details about these messages are described in the Oracle Fusion Applications Developer's Guide.

Message Properties

- The message type identifies the type of information that the message contains.
  See: Understanding Message Types

- The message name and number are identifiers for the message. There are specific message number ranges for predefined messages in each application, and you should not edit numbers assigned to predefined messages. When creating custom messages, use only message numbers within the 10,000,000 to 10,999,999 range.
  See: About Message Names
  See: About Message Numbers

- The translation notes for predefined messages might contain internal content that you can disregard.
  See: About Translation Notes

- The message category, severity, and logging enabled option are related to the incident and logging process.
  See: About Grouping Messages by Category and Severity
  See: Understanding Incidents and Diagnostic Logs with Message Dictionary

Message Text and Tokens

- The message text comprises various components, some of which are displayed only to select users. To determine which component of the message text is displayed to a particular user, set the Message Mode
profile option (FND_MESSAGE_MODE) at the user level for that user. The message component short text is visible to all users and therefore, the profile option does not apply to this component. Also, the profile option applies only to messages in the message dictionary.

See: About Message Components

- Tokens are variables that represent values to be displayed in the message text.

See: About Tokens

**Profile Options and Related General Preferences: How They Work Together**

Some Oracle Fusion Middleware Extensions for Applications profile options are related to general preferences in the global area.

### Preferences

The related general preferences are Default Application Language, Territory, Date Format, Time Format, Currency, and Time Zone. When the user changes any of these preferences, the stored values in LDAP are updated accordingly.

### Profile Options

The corresponding profile options are Default Language, Default Territory, Default Date Format, Default Time Format, Default Currency, and Default User Time Zone. No matter what you set for these profile options at any level, the preferences settings, or LDAP values, take precedence. The profile option value is used only if the LDAP value is not available. Updating the profile option value does not automatically update the value in LDAP or preferences.

**Document Sequences: Explained**

In Oracle Fusion Applications, each business document or business event is uniquely identified by a document sequence number that you assign to it. However, the document sequencing feature must be turned on (enabled) on the business document or event to allow the assignment. For example, if document sequencing is enabled, you can assign a document sequence number to an invoice that gets generated in response to a purchase order. You can use document sequences as a proof to track successfully executed transactions as well as failed transactions. Additionally, a document sequence helps in generating an audit trail, which can be used to identify how a particular transaction passed through various applications.

Document sequencing can be managed automatically, manually, and gaplessly.

**Note**

Plan your document sequencing carefully before you use the options available in the application to apply sequence numbers. Avoid changes to the options...
after you saved your work on the Manage Document Sequences and Manage Document Sequence Categories pages.

**Automatic Sequencing**

Automatic document sequencing assigns a unique number to each document as it is generated, and this unique number is stored in the database. The numbering is sequential by date and time of creation. If you define a sequence to automatically number documents, you can provide an initial value to begin the sequence. In absence of a custom value, the default value 1 is used.

**Manual Sequencing**

Manual sequencing requires you to assign a unique number to each document before it is generated. In manual sequencing, the numerical ordering and completeness of a transaction is not enforced. Users can skip or omit numbers when entering the sequence value. However, each time that a number is assigned, the application validates its uniqueness.

**Gapless Sequencing**

Gapless sequencing is similar to automatic sequencing. It automatically generates a unique number for each document, but does that only for successfully generated documents. As a result, the sequence is maintained for all the documents that are generated, and no sequence numbers are lost due to incomplete or failed document generation.

**Important**

Use this type of sequencing only if necessary because it may affect the performance of the system and slow down transaction processing.

**Sequential Numbering Enforced Profile Options**

The Sequential Numbering Enforced profile validates the documents being created and applies the relevant document sequence, based on the selected option. The following options are available:

- **Always Used**: Sequential numbering is enforced for all document categories. When this option is set, users cannot create a document if its document category is not assigned any active document sequence. The system displays an error.

- **Partially Used**: Sequential numbering is enforced for all document categories. However, if no sequence is available for the document, the system displays a warning. Users can either proceed without a document sequence or create and assign a document sequence before proceeding.

- **Not Used**: Sequential numbering is not enforced. In this mode, although the documents created do not require a document sequence to be assigned, the system checks for the uniqueness of the document number provided. The system does not display any warning or error.
At the site level, the profile is by default set to Partially Used. Avoid changing this option as doing so may interfere with the validation logic and the data that was stored using the earlier option may not appear. However, you can change the option at the product or user level.
Click to Dial: Explained

Use Click to Dial to place a call to a contact from a hyperlink on the phone number or phone icon.

Here are a few topics that are important to know when using Click to Dial:

- Normal call flow
- Interaction Records and Notes
- Operational Notes

Note

Click to Dial must be enabled to make calls using the various contact information pages and pop-up UIs. When enabled, phone numbers appear as hyperlinks. Interaction logging is available if that feature is enabled. If interaction logging is available, a note indicating that fact will be displayed in the UI. See Click to Dial: Top Tasks for more information about enabling these features.

Normal Call Flow

Click to Dial uses a call-the-caller-then-call-the-callee procedure for completing a phone call. That format and the normal flow of this procedure are described below.

- **You initiate a call**
  
  If you see a small orange square next to a contact or customer name, click the square to display further details, including phone numbers. To place a call, place your mouse over the phone number hyperlink and click.

Note

Click to Dial does not work on phone numbers that are marked with a Do Not Call icon.

- **Select a Calling Phone**
Choose the calling phone number. Usually the calling phone is a number from your profile information. Alternately, if you need to use a phone not in your profile, you can specify a different number to originate your call.

- Call Flow

After you select the calling phone number, the system calls you back on that number, waits for you to answer, and then calls the person for whom the call is intended.

Interaction Records and Notes

Click to Dial automatically creates an interaction record of the call, when that feature is enabled. The details window that provides the phone number may also show an Interaction icon that you can click to display a list of interaction records to edit, for example to provide a description of the call. The window may also provide a notes feature that you can use to record notes during the call.

Interaction Logging

The interaction record is logged as soon as the call is either successfully set up or known to have failed.

The interaction log records the customer, call participants, a timestamp noting the start time of the call, the direction of the communication, in or outbound, and
the resolution code. The description is automatically updated with these three items:

- Call ID from OWLCS
- Your chosen phone number
- Contact phone number

The call resolution code is determined from OWLCS and recorded in the interaction:

<table>
<thead>
<tr>
<th>OWLCS Call Status</th>
<th>Resolution Code in Interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>CallConnected</td>
<td>CALL ANSWERED</td>
</tr>
<tr>
<td>CallAborted</td>
<td>FAILED</td>
</tr>
<tr>
<td>CallHangUp</td>
<td>FAILED</td>
</tr>
<tr>
<td>CalledPartyBusy</td>
<td>NOT AVAILABLE</td>
</tr>
<tr>
<td>CalledPartyNoAnswer</td>
<td>NOT AVAILABLE</td>
</tr>
<tr>
<td>CalledPartyNotReachable</td>
<td>NOT AVAILABLE</td>
</tr>
<tr>
<td>CallingPartyBusy</td>
<td>FAILED</td>
</tr>
<tr>
<td>CallingPartyNoAnswer</td>
<td>FAILED</td>
</tr>
<tr>
<td>CallingPartyNotReachable</td>
<td>FAILED</td>
</tr>
</tbody>
</table>

Editing interactions

Once the call is established, if Interactions is available, you can use the Interactions icon on the UI to launch the interaction record list view. Select the current interaction record to edit it.

Operational Notes

Because of the call-the-caller-then-call-the-callee format, there are some conditions that may occur due to several calling situations. Some of these conditions are described below:

- Why don’t I hear a ring-back tone? As soon as you answer the system call-back, the system immediately dials the contact. You won’t hear a ring-back tone as in a normal outbound phone call. However, you can tell that the call attempt is progressing because:
  - The phone indicates that the connection is active. If the call to the contact reaches a busy tone or the call attempt times out, the connection is dropped.
  - The dialing window stays on the screen while the call attempt is progressing. It disappears when the connection is either successfully established or fails.

- What if your phone is busy and the call-back goes directly to voice mail? Normally this would not happen because you would not initiate a new call when you are already busy on another call. However, this situation could occur due to a race condition, that is where another incoming call reaches your phone before the Click to Dial call-back. When this happens, two different scenarios could occur:
• If your phone is configured for busy-forward-all-to-voice-mail, the Click to Dial call would be forwarded to your voice mail, and the system thinks that the caller has answered the call and will proceed to call the contact. On answering, the contact hears your voice-mail greeting.

• If your phone is capable of presenting a second call to the user, as is supported by many office phones and mobile phones, then you can still answer the Click to Dial call and there is no issue.

• What if you wait too long to answer the call-back? In other words, you wait longer than the ring-no-answer-forward-to-voice-mail timer on the phone system and the call goes to voice mail. Normally, this would not happen because you are expecting the inbound call after you started the call, and would answer promptly. However, if for some reason you do not answer and allow the call to ring-no-answer-forward to voice mail, then the system would think that you have answered the call and will proceed to call the contact. On answering, the contact hears your voice-mail greeting.

• What if the contact does not answer in 30 seconds and the system abandons the call attempt? If the contact’s voice mail is configured to answer after 30 seconds, you will not be able to leave a message.

Click-to-Dial: Top Tasks

Click-to-Dial is a feature of the customer contact process. Phone communication to customers and employees is initiated with a click of the mouse, leveraging your customer contact information and the application context. The Click-to-Dial feature uses Oracle WebLogic Communication Services, OWLCS, to enable communications. Applications that provide the Click-to-Dial functionality do so primarily through contextual actions in the application.

Additionally, Click-to-Dial utilizes Oracle Fusion Interactions as an optional transaction logging feature that will track information about the call such as the customer, call participants, a timestamp noting the start time of the call, the direction of the communication, in or outbound, and the resolution code.

Click-to-Dial integrates with your telephony environment and must be manually enabled in your deployment. This topic highlights what is required to set up the Click-to-Dial feature and to implement logging of the calls made using the Click-to-Dial feature.

Terms used in setting up these communications

• PSTN: Public switched telephone network is the network of the world’s public circuit-switched telephone networks.

• SIP: Session initiation protocol, an open signaling protocol standard that is used to set up phone calls

• TPCC: Third Party Call Control enables an application to control the telephony network to set up calls automatically.

• OWLCS: Oracle WebLogic Communication Services. Offers the TPCC service to Fusion applications and sets up the calls via SIP integration with the telephony network.
The set up task list Define WebLogic Communication Services Configuration delineates four tasks required for the correct configuration and implementation of Click-to-Dial. There is an optional task, separate from the set up task list, required for implementing Interaction logging.

Information about implementing Click-to-Dial can be found in the Oracle Fusion Applications Administrator's Guide. Detailed information about configuring and maintaining WebLogic Communication Services is found in the Oracle WebLogic Communication Services Administrator's Guide.

**Configure and Deploy WebLogic Server**

- Deploy WebLogic Communication Services: After the Oracle WebLogic communication server is deployed, this manual task activates the server.

  See: Oracle WebLogic Communication Services Administrator's Guide

**Integrate Communications Services**

- Integrate WebLogic Communication Services with Telephony Network: This manual task integrates communications within the telephony environment. OWLCS must be configured to interface with the specific characteristics of the telephony network.

  See: Managing Oracle WebLogic Communication Services for Click-to-Dial Functionality

**Specify the Domain and Address**

- Register a URL for the telephony gateway or soft switch for SIP domain: This task defines the Server protocol, defaulted to http, the external server host address and external server port address. The Associated Modules section is not required for setup. You can also perform this as a manual task using the Oracle Fusion Topology Manager to configure the address of the SIP Public Switched Telephone Network (PSTN) gateway or SIP soft switch serving the users within that domain. This address is needed by Click-to-Dial to correctly form the SIP addresses required by WebLogic Communication Services. See the link to Configuring PSTN Gateway Address Using Topology Manager: Worked Example.

**Enable Click-to-Dial**

- After configuring the server and defining the SIP domain, perform the Enable Click-to-Dial task. This task sets the value of the profile option Enable Click-to-Dial to 'Yes.'

**Call Logging via Fusion Interactions**

- To initiate the Interaction based logging for Click-to-Dial, set the profile option Call Interaction Logging Enabled to 'YES.'

---

**Configuring PSTN Gateway Address Using Topology Manager: Worked Example**

This example demonstrates how, during the implementation of the Register URL for the telephony gateway or soft switch for SIP domain task,
you must manually configure the PSTN gateway address by registering HzCTDPstnGatewayApp to a given environment using Oracle Fusion Topology Registration

These steps configure the address of the SIP Public Switched Telephone Network (PSTN) gateway or SIP soft switch serving the users within that domain. This address is needed by Click to Dial to correctly form the SIP addresses required by WebLogic Communication Services.

For example: SIP:+1650-555-1212@pstn_gateway.oracle.com;user=phone where pstn_gateway.oracle.com is the SIP domain. The SIP domain can also be expressed in the format 10.1.1.1 (IP address).

### Configuring PSTN using the Topology Manager

1. Sign in to Oracle Fusion Applications as a user that has application implementation consultant and WebLogic Services administration roles

2. In Fusion Applications Setup and Maintenance, click Register Enterprise Applications from the regional area under Topology Registration

3. On the Register Enterprise Applications page, click the plus icon to add an enterprise application. An Add Enterprise Application popup appears

4. Enter the new application information: Click Search in the Enterprise Application list field. Enter HzCTDPstnGatewayApp in the name field and click Search. Click OK.

5. Enter the other fields in the Add Enterprise Application popup

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>HzCTDPstnGatewayApp</td>
</tr>
<tr>
<td>Server Protocol</td>
<td>SIP</td>
</tr>
<tr>
<td></td>
<td>This field is ignored by click-to-dial. Oracle WebLogic Communication Service (OWLCS) always uses the SIP protocol.</td>
</tr>
<tr>
<td>External Server Host</td>
<td>10.143.167.91 (Used as an example)</td>
</tr>
<tr>
<td></td>
<td>A host name can be used instead of an IP address.</td>
</tr>
<tr>
<td>External Server Port</td>
<td>0 (Used as an example)</td>
</tr>
<tr>
<td></td>
<td>This field is ignored by click to dial.</td>
</tr>
</tbody>
</table>

6. Click Save and Close
Common Applications Configuration: Define Approval Management for Supply Chain Management

Approval Management: Highlights

Use approval management to determine the policies that apply to approval workflows for particular business objects such as expense reports. For example, you can specify levels of approval for expense reports over a particular amount, to reflect your own corporate policies. You also determine the groups of users who act on these workflow tasks, for example, the chain of approvers for expense reports.

Approval management is fully described in the Oracle Fusion Middleware Modeling and Implementation Guide for Oracle Business Process Management. Though the concepts described there apply also to Oracle Fusion Applications, the only setup relevant to Oracle Fusion Applications involves approval groups and task configuration. Customization of approval workflows themselves is described in the Oracle Fusion Applications Extensibility Guide.

Overview


  See: Introduction to Approval Management

  See: Understanding Approval Management Concepts

Approval Groups and Task Configuration

- An approval group consists of a name and a predefined set of users configured to act on a task in a certain pattern. Refer to the Oracle Fusion Middleware Modeling and Implementation Guide for Oracle Business Process Management.

  See: Administering Approval Groups

See: Using Task Configuration

• To configure a predefined approval policy, select the predefined rule set and click the Edit task icon button.

• To disable a predefined rule set, select the Ignore this participant check box for that rule set.

• To edit the rules within a predefined rule set, you can insert, update, or delete from the seeded rules as needed while in edit mode.

• You can configure a specific rule to automatically approve without being sent to any approver. Modify the routing for that rule so that it is sent to the initiator (which means the requestor is the approver), set the Auto Action Enabled option to True, and enter APPROVE in the Auto Action field.

Customization

• You can optionally customize predefined approval workflows, for example add post-approval activities or additional stages. Refer to the Oracle Fusion Applications Extensibility Guide.

See: Customizing and Extending SOA Components
Product Information Management Configuration: Define Basic Items

Manage Product Value Sets

Managing Product Value Sets: Explained

Value sets are specific to the application in which they will be used. In the Oracle Product Information Management application, value sets are used primarily for defining attributes where the values that an attribute can have is limited to a specific set of values.

Value sets can be edited or new value sets can be created from the Manage Product Value Sets page. The Edit icon launches the Edit Value Sets page, which redraws in the same region of the local area. The Create icon launches the Create Value Sets page, which redraws in the same region of the local area.

A value set is defined by the value set code and is specific to the module of an application in which the value set is to be used, such as Item Class.

The validation type determines how the value of field is validated for the assigned value set. The following are the seeded values:

- Format Only
- Independent
- Dependent
- Subset
- Table

The value data type determines the data type for the value set. The following are the seeded values:

- Character
- Number
- Date
• Date/Time

Manage Product Child Value Sets

Managing Product Child Value Sets: Explained

The Manage Product Child Value Sets task uses the same page as the Manage Product Value Set task.

A child value set is used to define variants for stock-keeping units or SKUs. A SKU contains the common properties for an item. For example, a shirt can be produced with colors; white, red, yellow, and blue. The variant is used to represent the colors of the shirt.

You define child value sets as follows:

• Create a value set with validation type of independent, for example All Colors.

• Select the new value set in the Manage Product Value Sets results table, for example All Colors.

• Click Manage Values, create several values, for example Blue, Red, Green, Yellow, and Black.

• Create a value set with validation type of Subset and enter the first value set you created for the independent value set, for example: Summer Colors.

• Select the value set Summer Colors in the Manage Product Value Set result table.

• Click Manage Values and then click the Add icon. The dialog will show a list of values based on the value set named Summer colors. Select two of them.

The value set Summer Colors is a child of All Colors.

Manage Default Item Class

Managing Default Item Class: Explained

The Root Item Class is seeded and all item classes are created as children of the Root Item Class. For Oracle Fusion Product Model customers, only the Root Item Class is available. The Manage Default Item Class task enables Product Model customers to manage item class templates, descriptive flexfields, attachment categories and lifecycle phases. The Manage Default Item Class task launches an edit page for the Root Item Class.

The functionality for the Root Item Class is defined using three tabs:
• The Basic tab enables descriptive flexfields and attachment categories to be viewed and managed for the Root Item Class.

• The Lifecycle Phases tab enables one or more lifecycle phases of a lifecycle to be associated with an item class.

• The Templates tab is where you define and manage item templates for the item class.

Manage Item Statuses

Managing Item Statuses: Explained

In the Item Status table, select a status code to display the associated attribute groups and attributes as well as control information.

You can create or edit or delete item statuses on the Manage Item Statuses page. Inactive dates are used to specify the date after which the item status will no longer be active. Operational attribute groups and attributes corresponding to the selected item status are displayed in the Details section. Select a value for the status from choice list for the attribute. Whenever the status is applied to the item, the value of the attribute may change. If the status will have no value, select No.

Select the Usage value of None or Defaulted or Inherited in the choice list for the Usage field that corresponds to how the attribute value will change based on the item status value:

- Defaulted-Sets the values of the item status attributes when the status value changes, but allows the overriding of the value during import and update of item.
- Inherited-Sets the values of the item status attributes when the status value changes, but overrides cannot occur.
- None-The item status attribute values will not be changed.

Any change made to an item status is not applied automatically to existing items, but will be applied during the editing of an item when the item status value is changed.

Manage Item Types

Managing Item Types: Explained

Item types are managed on the Manage Item Types page.

There are 32 seeded item types and you can edit them or create additional item types.

Item types are date-enabled and are made active or inactive by adjusting the Start Date and End Date.
To benefit from the use of item types, you must enable them by selecting the **Enable** checkbox.

### Manage Cross Reference Types

**Managing Cross-Reference Types: Explained**

Cross-References provide the functionality to map additional information about an item in the form of a value and cross-reference type. For example, the cross-reference can map between an item and an old part number, where the value is the value for the old part number and the type is Old Part Number. Cross-Reference Types are part of item relationships where the item relationship type is Cross-Reference. There are no values seeded for cross-reference types. You define the values using the **Manage Cross Reference Types** task. Cross-reference types are date-enabled and can be made active or inactive by adjusting the values of the Start Date and End Date. To benefit from using item relationship for cross-reference, you must enable cross-reference types by checking the **Enable** checkbox.

### Manage Item Descriptive Flexfields

**Managing Descriptive Flexfields for Items: Explained**

You can use descriptive flexfields to capture additional information about items beyond what is provided by the predefined set of operational attributes in Oracle Fusion Product Model.

**Item Descriptive Flexfields**

If you are not using Oracle Fusion Product and Catalog Management, then you cannot create user-defined attribute groups and attributes. However you can use descriptive flexfields associated at Item level to create fields to capture information about items. Like other descriptive flexfields, item descriptive flexfields have context segments and context-sensitive segments whose values are validated on entry by value sets. You can define the value sets to control what values users can enter in a descriptive flexfield segment. Examples of information that you might capture are size and volumetric weight.

Manage this flexfield type by using the **Manage Item Descriptive Flexfields** task, which you can access by searching for flexfield tasks on the Setup and Maintenance Overview page.

**Item Revision Descriptive Flexfields**

Use descriptive flexfields associated at Item Revision level to capture item revision information whose values may differ between revisions of the same item.
Manage this flexfield type by using the Manage Item Revision Descriptive Flexfields task, which you can access by searching for flexfield tasks on the Setup and Maintenance Overview page.

**Item Relationship Descriptive Flexfields**

When defining descriptive flexfields associated with item relationships, you must use certain prefixes when naming the context segments, in order for the segments to be displayed for the respective relationships.

The prefixes required for naming the context segments are listed in the following table, with their corresponding item relationship types. For example, if you define an item relationship descriptive flexfield with a context segment named `RELATED_RELATIONSHIP_ATTRIBUTES`, then the value segments of this context will be displayed for Related Item Relationships when users conduct transactions in that context. For another example, when users navigate to a UI of a particular object, such as a Competitor Item, they see the contexts whose internal name has the prefix `COMP`.

<table>
<thead>
<tr>
<th>Relationship Type</th>
<th>Prefix for Context Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Competitor Item Relationship</td>
<td>COMP</td>
</tr>
<tr>
<td>Customer Item Relationship</td>
<td>CUST</td>
</tr>
<tr>
<td>Item Cross-reference Relationship</td>
<td>XREF</td>
</tr>
<tr>
<td>GTIN Relationship</td>
<td>GTIN</td>
</tr>
<tr>
<td>Manufacturer Part Number Relationship</td>
<td>MFG</td>
</tr>
<tr>
<td>Related Item Relationship</td>
<td>RELATED</td>
</tr>
<tr>
<td>Source System Item Relationship</td>
<td>SYS</td>
</tr>
</tbody>
</table>

Manage this flexfield type by using the Manage Item Relationship Descriptive Flexfields task, which you can access by searching for flexfield tasks on the Setup and Maintenance Overview page.

**Trading Partner Item Descriptive Flexfields**

When defining descriptive flexfields associated with trading partner items, you must use certain prefixes when naming the context segments, in order for the segments to be displayed for the respective trading partner type.

The prefixes required for naming the context segments are listed in the following table, with their corresponding trading partner types. For example, if you define a trading partner item descriptive flexfield with a context segment named `COMP_TPI_ATTRIBUTES`, then the value segments of this context will be displayed for Competitor Item when users conduct transactions in that context.

<table>
<thead>
<tr>
<th>Trading Partner Type</th>
<th>Prefix for Context Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Competitor Item</td>
<td>COMP</td>
</tr>
<tr>
<td>Customer Item</td>
<td>CUST</td>
</tr>
<tr>
<td>Manufacturer Item</td>
<td>MFG</td>
</tr>
</tbody>
</table>
Manage this flexfield type by using the Manage Trading Partner Item Descriptive Flexfields task, which you can access by searching for flexfield tasks on the Setup and Maintenance Overview page.

**Import Items**

**Importing Items: Explained**

You can import items and item-related information using interface tables. This import data is loaded into the production tables using the Import Item task.

**Import Item**

The Import Item task creates an Enterprise Storage Server (ESS) process that takes the data that is loaded in the interface tables and uses the import process to move the data to the production tables. The import processes will perform all of the validations necessary to ensure the data imported is correct prior to moving the data into the production tables.

1. Access the Enterprise Storage Server and provide a process name (job definition) such as **Item Import Process**.
2. Select Setup and Maintenance from the Navigator.
   - Access the All Tasks tab on the Overview page, and search for the Import Item task with the name of your ESS process definition.
3. Click the **Go to Task** icon in the search results for that Import Item task.
   - The parameters for the item import process are
     - **Batch ID**: Associate the interface table to an item batch definition.
     - **Organization**: Select an organization to be used for the import.
     - **Process Only**: Determines how the data is processed. The choices are:
       - Create
       - Sync
       - Update
     - **Process All Organizations**: Select Yes if the import contains items that will be imported to multiple organizations.
     - **Delete Processed Rows**: Select Yes to delete rows that are imported without errors
4. Click **Submit** and the Request Number will be displayed.

**Monitoring Import Items**

Select Setup and Maintenance from the Navigator, then access Monitor Item Imports to search for specific Enterprise Storage Server processes and monitor their status in the search results table.
Manage Related Item Subtypes

Managing Related Item Subtypes: Explained

A related item is an item relationship between two existing items. How the two items are related is defined by a subtype. Multiple subtypes for related items are seeded, and you define additional subtypes on the Manage Related Item Subtypes page.

FAQs for Define Basic Items

What's the difference between lifecycle phase types and lifecycle phases?

Lifecycle phase types are seeded and describe the type of lifecycle phase. They are Design, Obsolete, Preproduction or Prototype, and Production.

Lifecycle phases must be created by the user by selecting one of the seeded lifecycle phase types.
Cost Management Configuration: Define Cost Management

Manage Cost Organizations, Cost Organization Relationships, and Cost Books

Cost Organizations, Inventory Organizations, and Cost Books: How They Fit Together

A cost organization structure comprises cost organizations, inventory organizations, and cost books. Your accounting and business needs determine how you set up your cost organization structure. This structure in turn determines how the cost processors create cost accounting distributions and accounting entries for inventory transactions.

This figure illustrates the relationship between cost organizations, inventory organizations, and cost books.
A cost organization can represent a single inventory organization, or a group of inventory organizations that roll up to a business unit. You can group several inventory organizations under a cost organization for financial reporting purposes. However, a cost organization can map to only one business unit.

The inventory organizations that are assigned to a cost organization must all belong to the same legal entity.

For each cost organization, define an item validation organization from which the processor should derive the default units of measure. You can designate one of the inventory organizations assigned to the cost organization to be the item validation organization, or you can designate the item master organization to be the item validation organization.

Cost Books

A cost book sets the framework within which accounting policies for items can be defined. You can define different cost books for each of your financial accounting, management reporting, and analysis needs. By assigning multiple cost books to a cost organization, you can calculate costs using different rules simultaneously, based on the same set of transactions.

Every cost organization must have one primary cost book that is associated with the primary ledger of the legal entity to which the cost organization belongs. You can also assign secondary ledger-based cost books for other accounting needs,
as well as ledgerless cost books for simulation purposes. For example, you could assign a primary cost book for financial reporting, a secondary cost book for business analysis, and a ledgerless cost book to simulate results using different cost calculations.

When you assign a cost book to a cost organization, you can optionally associate it with a ledger. The cost book then inherits the currency, conversion rate, cost accounting periods, and period end validations of that ledger. If you are assigning a ledgerless cost book, then you define these elements manually.

**Setting Up the Cost Organization Structure: Points to Consider**

Set up your cost organization structure to accommodate your costing and accounting needs. The following discusses considerations for creating cost organizations, their association with inventory organizations, and their assignment to cost books.

**Creating Cost Organizations**

When deciding what cost organizations to set up, consider the following:

- Financial reporting. You typically create a separate cost organization for every business unit.
- Data security needs. The cost organizations that you create may be determined by the separation of duties and security requirements for your users.

**Note**

You can use the effective start and end dates on the Manage Cost Organization Relationships page, to manage changes in the existence of a cost organization, for example, due to mergers, acquisitions, or restructuring.

**Using Cost Organization Sets**

By assigning cost organizations to a set, the entities defined at the set level can be shared by all the cost organizations belonging to that set. A cost organization set enables you to streamline the setup process, and helps you avoid redundant setup by sharing set-level definitions of your cost profiles, valuation structures, cost elements, and cost component groups across the cost organizations that belong to the set.

You also have the flexibility to assign cost organizations to different sets, for example if they are in different lines of business. That way you can segregate the definitions that are shared.

**Associating Inventory Organizations with Cost Organizations**

Your operation may lend itself to a simple configuration of one inventory organization to one cost organization. Or, when there are many inventory organizations in the same business unit, you may group several inventory organizations under a single cost organization for any of the following reasons:
• Costing responsibilities. You may want to group inventory organizations that roll up to a manager or a cost accounting department under the same cost organization.

• Uniform cost accounting. For example, if you want to define your overhead rules just once and apply them to transactions from several inventory organizations, you could group those inventory organizations into one cost organization.

• Cost sharing. If there are items in more than one inventory organization for which you want a single average cost, those inventory organizations must fall under the same cost organization.

---

**Assigning Cost Books to Cost Organizations**

Every cost organization must be assigned one primary cost book that is associated with the primary ledger of the legal entity to which the cost organization belongs. You may also assign several secondary cost books as needed for other purposes such as: business analysis and management reporting, local currency accounting, or profit tracking of inventory items.

You can also assign ledgerless cost books to a cost organization for simulation purposes.

---

**Setting Up a Cost Organization Structure: Examples**

The following examples illustrate cost organization structures that support different cost accounting needs.

**Example 1**

Set up three inventory organizations to optimize materials management across three different locations. Because they all belong in the same business unit and are managed by one cost accounting department, you could group them under a single cost organization; or you could assign each inventory organization to its own cost organization.

**Example 2**

Three inventory organizations are geographically dispersed, and each one falls under a separate business unit. Create three cost organizations, and assign each inventory organization to its own cost organization.

**Example 3**

Four inventory organizations are geographically dispersed. Two of them fall under one business unit, and two fall under another business unit. You could group the inventory organizations under two cost organizations corresponding to the two business units; or you could assign each inventory organization to its own cost organization.
Example 4

Two inventory organizations in the same business unit need to share a single average cost for some items. These inventory organizations must belong to the same cost organization.

FAQs for Manage Cost Organizations and Cost Organization Relationships

What's a set-level definition?

A set-level definition enables you to segment and share your reference data. Entities that are defined at the set level can be shared by all cost organizations belonging to that set. For example, to segment your cost element reference data by country, you can define cost elements for each country set; and the cost organizations belonging to the country set can share the cost elements within that set. You can also use the Common set to share the same reference data across all cost organizations. This saves you redundant setup, and streamlines the process.

Can I change the legal entity of a cost organization?

No. You cannot change the legal entity of a cost organization once transactions are processed under that cost organization.

How can I create and maintain a cost organization?

You can create, edit, or delete a cost organization in the Oracle Fusion Global Human Resources application, on the Manage Cost Organization page.

Can I associate an inventory organization with more than one cost organization?

No. You can associate an inventory organization with only one cost organization.

Can I delete or inactivate a cost book or a cost book assignment to a cost organization?

Yes. You can delete or inactivate a cost book or a cost book assignment to a cost organization if there are no costing transactions or other references that depend on the cost book or cost book assignment. Do this by first deleting references to the cost book in other cost management setup, then delete the cost book. Likewise, first delete references to the cost book assignment in other cost management setup, then delete the cost book assignment.

You can inactivate a cost book or cost book assignment to a cost organization at any time. To inactivate a cost book or cost book assignment, set the effective end date to a current or future date; however, all past assignments remain in effect.

Can I delete or inactivate the association of an inventory organization with a cost organization?

Yes. You can delete or inactivate the association of an inventory organization with a cost organization, but only if there are no costing transactions or other
references that depend on the inventory organization and cost organization relationship. Do this by first deleting all references to the inventory organization and cost organization association in other cost management setup, then delete the association.

You can also inactivate the association of an inventory organization with a cost organization by setting the effective end date to a current or future date; however, all past associations remain in effect.

Manage Cost Analysis Groups

Cost Elements, Analysis Codes, and Analysis Groups: Explained

Map cost elements to analysis codes within analysis groups for reporting purposes. This enables you to categorize and summarize your costs differently, depending on your reporting needs.

Use analysis groups to define alternate views of item costs. For example, you can group cost elements into a fixed and variable analysis group, or a direct cost and indirect cost analysis group.

Using Cost Elements, Analysis Codes, and Analysis Groups

You can assign a cost element to multiple analysis codes. However an analysis code must be unique within an analysis group, and it can be reused in multiple analysis groups. For each analysis group you can set up a default analysis code that is used for unassigned cost elements.

The following are examples of cost elements classified under analysis groups.

<table>
<thead>
<tr>
<th>Analysis Group</th>
<th>Analysis Code</th>
<th>Cost Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable/Fixed Costs</td>
<td>Variable Cost</td>
<td>Direct Material</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inbound Freight</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Material Handling</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Outbound Freight</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Direct Labor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Internal Profits</td>
</tr>
<tr>
<td></td>
<td>Fixed Cost</td>
<td>Store Supervisor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Factory Rent</td>
</tr>
<tr>
<td>Indirect/Direct Costs</td>
<td>Indirect Cost</td>
<td>Outbound Freight</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Internal Profits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Store Supervisor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Factory Rent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Electricity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Depreciation</td>
</tr>
<tr>
<td>Direct Cost</td>
<td></td>
<td>Direct Material</td>
</tr>
</tbody>
</table>
Cost components come from external sources, and are mapped to cost elements which the costing application uses to track the cost of items. Use cost component groups to map cost components to cost elements, and to map source cost elements to destination cost elements when items are transferred from one inventory organization to another.

This figure illustrates the relationship between cost components, cost elements, cost component groups, and cost profiles.

Cost components are the most granular representation of item costs. They come into the costing application from external sources such as inventory, receiving, and accounts payable. Examples of cost components are purchase order item price, material, freight, and tax.

Cost Elements

A cost element is the level where the costs of an item are tracked through the inventory accounting life cycle. Cost components are mapped to cost elements, which enables you to calculate item costs at different granularity levels for different business needs. For example, you may want more granularity for high-value than for low-value items.

You can define cost elements for four types of costs:
• Material cost element type for incoming material cost components.

• Overhead cost element type for costs that are calculated by the cost processor based on user-defined overhead rules.

• Profit in Inventory cost element type for tracking of internal margins when items are transferred from one inventory organization to another, including global procurement and drop shipment flows. For cost elements of this type, indicate the Profit in Inventory organization that incurs the gain or loss due to the transfer of goods.

• Adjustment cost element type for separate tracking of cost adjustments, which provides a more detailed view of item costs and profit margins.

Cost elements are defined at the set level and thereby have the advantages of set-level definitions for sharing and segregation. A Profit in Inventory cost element must be assigned to the Common cost element set so that it can be shared across cost organizations.

The following are examples of cost element definitions:

<table>
<thead>
<tr>
<th>Cost Element Set</th>
<th>Cost Element</th>
<th>Cost Element Type</th>
<th>Inventory Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>Metals Material</td>
<td>Material</td>
<td></td>
</tr>
<tr>
<td>US</td>
<td>Plastic Material</td>
<td>Material</td>
<td></td>
</tr>
<tr>
<td>US</td>
<td>Misc Material</td>
<td>Material</td>
<td></td>
</tr>
<tr>
<td>US</td>
<td>Misc Material</td>
<td>Adjustment</td>
<td></td>
</tr>
<tr>
<td>US</td>
<td>Plant Depreciation</td>
<td>Overhead</td>
<td></td>
</tr>
<tr>
<td>US</td>
<td>Equipment Depreciation</td>
<td>Overhead</td>
<td></td>
</tr>
<tr>
<td>US</td>
<td>Freight Charges</td>
<td>Overhead</td>
<td></td>
</tr>
<tr>
<td>Common</td>
<td>Internal Margin</td>
<td>Profit in Inventory</td>
<td>Seattle</td>
</tr>
<tr>
<td>UK</td>
<td>Dairy Material</td>
<td>Material</td>
<td></td>
</tr>
<tr>
<td>UK</td>
<td>Misc Material</td>
<td>Material</td>
<td></td>
</tr>
<tr>
<td>UK</td>
<td>Dairy Material</td>
<td>Adjustment</td>
<td></td>
</tr>
</tbody>
</table>

Cost Component Groups

Use cost component groups to define mappings of:

• Cost components from external sources to cost elements in the costing application. These mappings provide flexibility in the granularity level where you track costs. You can map one cost component to one cost element for a detailed cost breakdown, or several cost components to one cost element for a less granular view of costs.

• Source cost elements to destination cost elements in the transfer of items from one inventory organization to another. This helps to maintain visibility of the item cost structure from the source application and across the supply chain.

You can specify a default cost component mapping to cost element to be used in cases where the source cost element does not have a matching destination cost element. The default cost component mapping is helpful when:
- The detailed mapping of a cost component to cost element is not required, and you want to map it to a single cost element.
- The designated mapping for a cost component is missing. If the mapping is missing, the transaction automatically picks up the default cost component mapping.

**Note**

If the cost component mapping is missing, the cost processor logs a message in the processing log. If the cost component mapping is missing and there is no default mapping, you can create the mapping and the transactions will be processed in the next run. If there is a default mapping, the transaction is processed and you can review the message log to decide if you want to take further action: you can correct the mapping for future transactions, and you can create a cost adjustment to reclassify the costs as needed.

Cost component groups are one of the attributes of cost profiles, which the cost processor uses to determine how to calculate item costs. Cost component groups are defined at the set level and thereby have the advantages of set-level definitions for sharing and segregation. Cost component groups and cost profiles are both set enabled; therefore, only those cost component groups belonging to the same set as the cost profile are available to that cost profile.

The following are examples of cost component group mappings.

**Example 1:** Mapping of one cost component to one cost element.

<table>
<thead>
<tr>
<th>Mapping Group</th>
<th>Cost Component</th>
<th>Cost Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>MG1</td>
<td>PO Item Price</td>
<td>Material</td>
</tr>
<tr>
<td></td>
<td>PO Tax</td>
<td>Tax</td>
</tr>
<tr>
<td></td>
<td>Profit in Inventory</td>
<td>PII</td>
</tr>
<tr>
<td></td>
<td>Interorganization Freight</td>
<td>Freight Charges</td>
</tr>
<tr>
<td></td>
<td>Invoice Price Variance</td>
<td>IPV</td>
</tr>
<tr>
<td></td>
<td>Exchange Rate Variance</td>
<td>ERV</td>
</tr>
<tr>
<td></td>
<td>Tax Invoice Price Variance</td>
<td>TIPV</td>
</tr>
</tbody>
</table>

**Example 2:** Cost components mapped to one or more cost elements.

<table>
<thead>
<tr>
<th>Mapping Group</th>
<th>Cost Component</th>
<th>Cost Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>MG2</td>
<td>PO Item Price</td>
<td>Material</td>
</tr>
<tr>
<td></td>
<td>PO Tax</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NR Tax</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Invoice Price Variance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Exchange Rate Variance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tax Invoice Price Variance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Interorganization Freight</td>
<td>Freight Charges</td>
</tr>
<tr>
<td></td>
<td>Profit in Inventory</td>
<td>PII</td>
</tr>
</tbody>
</table>
Example 3: Mapping of source cost elements to destination cost elements in an interorganization transfer.

<table>
<thead>
<tr>
<th>Mapping Group</th>
<th>Source Cost Element Set</th>
<th>Source Cost Element</th>
<th>Destination Cost Element Set</th>
<th>Destination Cost Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>MG3</td>
<td>US</td>
<td>Material Tax</td>
<td>UK</td>
<td>Material</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Freight Charges</td>
<td></td>
<td>Freight Charges</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other</td>
<td></td>
<td>Other</td>
</tr>
</tbody>
</table>

You have flexibility in how you map cost component groups to items:

- Different items in a cost organization and book combination can have the same or different cost component group mappings if they use different cost profiles.
- One item can have different cost component group mappings in different cost books.
- Several cost organizations can share the same cost component group mappings if they belong to the same set, or if they are defined the same way in different sets.

The following figure illustrates different mappings of cost component groups to items.

**Cost Profiles, Default Cost Profiles, and Item Cost Profiles: Explained**

Cost profiles define the cost accounting policies for items. The cost processor refers to the attributes of a cost profile to calculate costs and create accounting.
distributions for item transactions. Each item in a cost organization book requires a cost profile to calculate the inventory transaction costs.

The following describes how to define cost profiles and assign them to items.

**Cost Profile Definition**

A cost profile has the following attributes:

- **Cost profile set.** Cost profiles use set-level definitions, and all cost organizations belonging to that set can share the same cost profile definitions.

- **Cost method.** Establishes how cost is calculated. The costing application uses the perpetual average cost or actual cost method.

- **Valuation structure.** Sets the granularity level at which items are costed, for example Cost Organization level, or Inventory Organization, Lot, and Grade level.

- **Valuation structure type.** Asset or Expense type.

- **Cost component group.** Maps incoming cost components to cost elements, which are used to cost transactions.

- **Costing unit of measure (UOM).** Used to cost the item. Some items have primary and secondary units of measure, and there is no fixed conversion factor between the two. For example, you can calculate the cost of chickens by eaches or by weight.

- **Quantity depletion method.** Sets how inventory quantity is depleted when inventory transactions are costed. The method used by the costing application is first in, first out (FIFO).

- **Method for processing negative quantity.** Establishes how to treat depletion of inventory when the depletion quantity exceeds inventory on hand. Options are Always, To Zero, or Never.

- **Propagate cost adjustment.** Option to propagate cost adjustments down the supply chain, available only if using the actual cost method.

- **Receipt without cost.** Specifies that for sales returns without an RMA reference or missing incoming cost, the cost processor uses the first or last receipt layer cost.

- **Referenced RMA cost.** Specifies that for sales returns with an RMA reference, the cost processor uses the average cost of the original sales issue.

**Cost Profile Assignment to Items**

Assign cost profiles to items in each cost organization and book combination where there are item transactions. Cost organizations can have multiple cost books and the same item can have different cost profiles in different cost books used by the cost organization. This is useful when you want to use different books for various financial reporting and decision making purposes, such as statutory reporting, or management reporting.
Items can also use different cost profiles in various cost organization and book combinations, when they require different cost accounting policies.

Note

An item can have only one asset and one expense cost profile in each cost organization and book combination.

You can simplify the effort by assigning default cost profiles at the cost organization book level, or at the item category level within the cost organization book. Default cost profiles are generally used if the costing policy is the same for all items in the cost organization book, or in the item category.

Override default cost profiles by assigning specific item cost profiles at the individual item level. You can modify or delete a default cost profile assignment at any time, and these changes will be in effect for subsequent transactions.

You can assign cost profiles in three ways:

- **Automatic without approval.** If the default cost profile has the new item profile creation mode set to Auto, the preprocessor automatically generates and assigns the default cost profile to new items. This means that the cost processor uses the same cost profile for all items within that cost organization book, or within the item category.

- **Automatic with approval.** If the default cost profile has the new item profile creation mode set to Review Required, you must review and approve the generated cost profile before the cost processor assigns it to new items.

- **Manual.** Manually assign the cost profile to a new item before the cost processor processes the first transaction. This cost profile then remains in effect for subsequent transactions. The manually assigned cost profile always takes precedence over the defaulted cost profile.

Note

When you are manually assigning a cost profile to an item, the options available are both the cost profiles belonging to the set that is specific to the cost organization of the item, and the profiles belonging to the Common set, which spans all cost organizations.

**New Item Cost Profile Creation: Review and Approval Process**

One of the attributes of a default cost profile is the new item cost profile creation mode, which can be set to Auto or Review Required. The creation mode determines how a new item cost profile is created.

**Settings That Affect the Review and Approval Process**

If a default cost profile has the new item profile creation mode set to Auto, the preprocessor automatically assigns cost profiles to new items, and they can be immediately used by the cost processor. If a default cost profile has the new
item profile creation mode set to Review Required, then you must review and approve cost profiles before they can be used.

**How a New Item Cost Profile Is Reviewed and Approved**

When you save a default cost profile that is in Review Required mode, the preprocessor generates the cost profiles for new items based on the default cost profile, and sets them to Awaiting Approval status.

You then review the cost profiles on the Review and Approve Item Cost Profiles page, which is accessed from the Manage Default Cost Profiles page, or from the Cost Accountant Dashboard. After reviewing the cost profiles that are in Awaiting Approval status, set them to Approved or Rejected status. If you approve them, the creation source becomes Default Cost Profile. If you reject the cost profiles, you can manually modify them, and the creation source becomes Manual.

Save your changes and rerun the preprocessor, for final assignment of the item cost profiles.

**Valuation Structures and Valuation Units: Explained**

Valuation structures and valuation units define the granularity level at which the cost of an item is maintained. For example, you can maintain your cost calculations at the lot ID, serial ID, grade, or item level. Under certain circumstances, you can even have a single average cost for an item spanning more than one inventory organization.

The following describes how to use valuation structures and valuation units.

**Valuation Structures**

A valuation structure defines the level at which item costs are maintained. It specifies which inventory control attributes are used to segregate costs, and it is one of the attributes of an item cost profile. When a cost profile is assigned to an item, the cost processor uses the valuation structure on that cost profile to determine how to calculate the item cost.

The flexfield structure specifies the costing attributes that are enabled for a valuation structure. The costing attributes can be one or more of the following: Cost Organization (mandatory), Inventory Organization, Subinventory, Locator, Lot, Serial, and Grade. The costing attributes must be consistent with the inventory attributes, and cannot be at a lower level of granularity than the inventory on hand.

Valuation structures are of two types, asset and expense. An asset valuation structure is used for receipts of items that are valued as inventory on the balance sheet. An expense valuation structure is used to account for receipts to inventory of items that are expensed rather than treated as assets on the balance sheet. A cost profile with an asset valuation structure becomes an asset cost profile, and a cost profile with an expense valuation structure becomes an expense cost profile. Define both the asset and expense valuation structures on your cost profile; the item then inherits both valuation structures when it is associated with the cost profile.
Valuation structures are defined at the set level and thereby have the advantages of set-level definitions for sharing and segregation.

The valuation structure mode determines whether the valuation units are created manually, or automatically by the cost processor, or both:

- Valuation structures that you define as automatic are those that tend to be unlimited and unknowable in advance, such as lot IDs and serial IDs. With the automatic setting, the cost processor automatically creates a new valuation unit code as transactions for new lot IDs or serial IDs are processed.

- Valuation structures that you define as manual are those that tend to have a finite list of possibilities, such as grades or subinventories. Use the manual mode when you want to ensure that transactions that do not meet one of the expected possibilities will trigger an error condition.

- Valuation structures that you set as both manual and automatic are those cases where you can either define the anticipated valuation units before they enter the processor, or you can let the processor automatically create the valuation units if you have not already created them manually.

Valuation Units

A valuation unit is the set of values for the control attributes defined by the valuation structure. For example, valuation unit V1 comprises cost organization A, lot L1, and grade G1, and valuation unit V2 comprises cost organization B, lot L2, and grade G2. The processor calculates two different costs for the item: a cost for valuation unit V1 and a cost for valuation unit V2.

You can define multiple valuation units under a valuation structure, using different combinations of these costing attributes. The cost processor will automatically generate these valuation units if the valuation structure mode is set to Auto or Both.

By assigning a valuation unit to a cost organization book you specify the set of values for the inventory control attributes that are used to cost the item within that cost organization book.

Valuation Structure Rules: Explained

The valuation structure is one of the attributes of an item cost profile which is used to cost inventory items.

Conflicts may arise if the inventory control attributes in the valuation structure do not match the inventory control attributes of the inventory items.

Valuation Structure Conflict Resolution

In cases where the valuation structure specifies an inventory control attribute that is missing on the item, the cost processor applies the following rules.

If the inventory control attribute has the Required attribute set to Yes, then the association of the valuation structure to the item is disallowed. If the inventory control attribute has the Required attribute set to No, then the association of the
valuation structure is allowed, and the valuation unit will have a Null value for that inventory control attribute.

For example, suppose an item is not lot enabled, whereas lot is an attribute of the valuation structure. In this case, if the Required attribute is set to No, the valuation structure is considered valid for the item, and the processor applies Null to the lot value. However, if the Required attribute is set to Yes, the valuation structure is considered invalid.

The Required attribute can be changed from Yes to No at any time to accommodate missing values. However it cannot be changed from No to Yes if any transactions using that valuation structure have been processed.

**Using Valuation Structures to Calculate Item Costs: Examples**

The following are examples of how the cost accounting application maintains costs for an item using different valuation structures.

Assume that a cost organization stocks an item in four stores under two inventory organizations, as follows.

<table>
<thead>
<tr>
<th>Cost Organization</th>
<th>Inventory Organization</th>
<th>Subinventory</th>
<th>Lot</th>
<th>Item</th>
<th>Quantity</th>
<th>Cost per Unit</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO-US</td>
<td>Retail Store 1</td>
<td>Store 1A</td>
<td>A</td>
<td>Gadget A</td>
<td>50</td>
<td>50</td>
<td>2500</td>
</tr>
<tr>
<td>CO-US</td>
<td>Retail Store 2</td>
<td>Store 2B</td>
<td>C</td>
<td>Gadget A</td>
<td>45</td>
<td>44</td>
<td>1980</td>
</tr>
<tr>
<td>CO-US</td>
<td>Retail Store 2</td>
<td>Store 2A</td>
<td>C</td>
<td>Gadget A</td>
<td>60</td>
<td>45</td>
<td>2700</td>
</tr>
<tr>
<td>CO-US</td>
<td>Retail Store 1</td>
<td>Store 1B</td>
<td>B</td>
<td>Gadget A</td>
<td>40</td>
<td>48</td>
<td>1920</td>
</tr>
</tbody>
</table>

**Example 1**

The application calculates and maintains the item cost at the cost organization level. It maintains one cost for the item across all inventory organizations in the cost organization.

<table>
<thead>
<tr>
<th>Valuation Structure</th>
<th>Unit Average Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost Organization</td>
<td>46.67</td>
</tr>
</tbody>
</table>

**Example 2**

The application calculates and maintains the item cost separately for each inventory organization in the cost organization.

<table>
<thead>
<tr>
<th>Valuation Structure</th>
<th>Inventory Organization</th>
<th>Unit Average Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost Organization - Inventory Organization</td>
<td>Retail Store 1</td>
<td>49.11</td>
</tr>
<tr>
<td>Cost Organization - Inventory Organization</td>
<td>Retail Store 2</td>
<td>44.57</td>
</tr>
</tbody>
</table>
Example 3

The application calculates and maintains the item cost separately for each subinventory in the cost organization.

<table>
<thead>
<tr>
<th>Valuation Structure</th>
<th>Inventory Organization</th>
<th>Subinventory</th>
<th>Unit Average Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inventory Organization - Subinventory</td>
<td>Retail Store 1</td>
<td>Store 1A</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Retail Store 1</td>
<td>Store 1B</td>
<td>48</td>
</tr>
<tr>
<td>Inventory Organization - Subinventory</td>
<td>Retail Store 2</td>
<td>Store 2A</td>
<td>45</td>
</tr>
<tr>
<td>Inventory Organization - Subinventory</td>
<td>Retail Store 2</td>
<td>Store 2B</td>
<td>44</td>
</tr>
</tbody>
</table>

Example 4

The application calculates and maintains the item cost separately for each lot under each subinventory and inventory organization.

<table>
<thead>
<tr>
<th>Valuation Structure</th>
<th>Inventory Organization</th>
<th>Subinventory</th>
<th>Lot</th>
<th>Unit Average Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inventory Organization - Subinventory - Lot</td>
<td>Retail Store 1</td>
<td>Store 1A</td>
<td>A</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Retail Store 1</td>
<td>Store 1B</td>
<td>B</td>
<td>48</td>
</tr>
<tr>
<td>Inventory Organization - Subinventory - Lot</td>
<td>Retail Store 2</td>
<td>Store 2A</td>
<td>C</td>
<td>45</td>
</tr>
<tr>
<td>Inventory Organization - Subinventory - Lot</td>
<td>Retail Store 2</td>
<td>Store 2B</td>
<td>C</td>
<td>44</td>
</tr>
</tbody>
</table>

Example 5

The application calculates and maintains the item cost separately for each lot under each inventory organization.

<table>
<thead>
<tr>
<th>Valuation Structure</th>
<th>Inventory Organization</th>
<th>Subinventory</th>
<th>Lot</th>
<th>Unit Average Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inventory Organization - Lot</td>
<td>Retail Store 1</td>
<td>Store 1A</td>
<td>A</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Retail Store 1</td>
<td>Store 1B</td>
<td>B</td>
<td>48</td>
</tr>
<tr>
<td>Inventory Organization - Lot</td>
<td>Retail Store 2</td>
<td>Store 2A</td>
<td>C</td>
<td>44.57</td>
</tr>
<tr>
<td>Inventory Organization - Lot</td>
<td>Retail Store 2</td>
<td>Store 2B</td>
<td>C</td>
<td>44.57</td>
</tr>
</tbody>
</table>
Manage Costing Units of Measure

Costing Units of Measure: Explained

You can cost an item using different units of measure (UOMs) for different business purposes, such as pricing, reporting or tracking costs.

The UOM is one of the attributes of a cost profile. You can calculate different costs for an item by assigning it cost profiles with different UOMs.

Primary and Secondary UOMs

To illustrate the use of a primary or secondary UOM, consider the case of chickens that can be costed by a UOM of each or of pounds. There is no standard conversion from one UOM to the other. In such a case, the costing UOM depends on how the chickens are sold, priced, or tracked. It may be more logical to cost the chickens by pound if that is how they are sold. However it may be more useful to cost them by each for planning and tracking purposes. In this case, the primary UOM could be each, and the secondary UOM could be pounds.

When an item in a cost organization and book combination is assigned a cost profile that specifies the use of the primary or secondary UOM, the cost accounting application uses the primary or secondary UOM that is defined in the item validation organization.

Using Different Units of Measure to Cost an Item: Example

This example illustrates how to calculate costs for an item using different units of measure.

Scenario

Consider a jewelry retail business that sells gold rings. The company purchases the rings in dozens, and maintains inventory costs in dozens and eaches.

Shipment Quantities and Costs

The company receives five shipments of rings as follows.

<table>
<thead>
<tr>
<th>Shipment Number</th>
<th>Number of Rings</th>
<th>Total Shipment Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2 dozen</td>
<td>$4,800</td>
</tr>
<tr>
<td>2</td>
<td>3 dozen</td>
<td>$5,400</td>
</tr>
<tr>
<td>3</td>
<td>5 dozen</td>
<td>$10,500</td>
</tr>
<tr>
<td>4</td>
<td>2 dozen</td>
<td>$5,400</td>
</tr>
<tr>
<td>5</td>
<td>6 dozen</td>
<td>$7,200</td>
</tr>
</tbody>
</table>

Analysis

Define a primary unit of measure (UOM) of dozens and a secondary UOM of eaches. You can calculate two different costs for each shipment using the primary UOM and the secondary UOM.
Resulting Costs in Primary and Secondary UOMs

The costs using the primary versus the secondary UOMs are as follows.

<table>
<thead>
<tr>
<th>Shipment Number</th>
<th>Cost in Primary UOM</th>
<th>Cost in Secondary UOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2,400</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>$1,800</td>
<td>150</td>
</tr>
<tr>
<td>3</td>
<td>$2,100</td>
<td>175</td>
</tr>
<tr>
<td>4</td>
<td>$2,700</td>
<td>225</td>
</tr>
<tr>
<td>5</td>
<td>$1,200</td>
<td>100</td>
</tr>
</tbody>
</table>

FAQs for Define Costs and Cost Book Policies

Can I delete or modify a cost profile, a default cost profile, or an item cost profile?

No. You cannot delete or modify a cost profile after it has been used to cost transactions for an item. However, if a cost profile has not been used to cost any transactions, you can delete or modify it after you delete references to it in other cost management setup.

What happens when I select different options for processing negative inventory quantities?

The options for processing inventory quantities when the transaction quantity exceeds the quantity on hand are:

- Always: applies cost for the entire transaction, including negative balances. The cost processor costs the transaction as follows:
  - If the cost method is perpetual average cost, it applies the average cost for the entire transaction quantity.
  - If the cost method is actual cost, it applies the FIFO layer cost for the entire transaction quantity, and then processes a cost variance when the next receipt replenishes inventory.
- To Zero: applies cost only for quantity on hand, and holds the remaining shortfall until inventory is replenished.
- Never: does not apply cost for the transaction until quantity is sufficient to cover the entire transaction.

Define Overheads

Expense Pools, Cost Element Groups, and Overhead Accounting Rules: How They Fit Together

Use expense pools, cost element groups, and overhead accounting rules to calculate overhead absorption for inventory transactions. Overhead expenses
can be absorbed and capitalized into inventory, or they can be absorbed and reclassified as an expense.

**Overhead Costs Expensed or Capitalized**

On inbound transactions and inventory transfer transactions, overhead expenses can be absorbed and capitalized into inventory value, or the absorption can be redirected to an expense account: a credit to an absorption account and a debit to either an inventory or expense account. On outbound transactions, overhead absorption is redirected to an expense account, and will be included in the gross margin calculation.

For example, consider a receipt of inventory items that cost $10 each to purchase, and you would like to absorb overhead cost of $2 each on the inbound transaction. When the item is sold, you would like to absorb additional overhead of $3 each on the outbound transaction. The total cost of goods sold is $15 each.

**Expense Pools**

Expense pools represent a collection of general ledger expense accounts that can be absorbed as overhead costs. Expense pools are defined at the cost organization level. Overhead rules are defined for expense pools, and an expense pool can have many overhead rules that absorb it.

Expense pools are mapped to a cost element, and a cost element can contain one or more expense pools. When overhead is absorbed, an accounting distribution is created for each expense pool, so you can define accounting rules crediting the absorption account at the expense pool level. Once the inbound transaction is in inventory, the application tracks the value of inventory at the cost element level, so that you can track costs through inventory at the desired level of granularity.

**Cost Element Groups**

Cost element groups tell the processor which cost elements to sum when the overhead rule is a percentage of cost. Cost element groups can be system defined or user defined, and they are set at the cost organization level.

There are two predefined cost element groups, Transaction Cost and Material. You can also define your own cost element groups.

**Overhead Accounting Rules**

The application uses the overhead accounting rules that you define to determine when and how overhead costs should be calculated. Overhead calculations are based on cost element pools or cost element groups.

Overhead accounting rules are defined at the cost organization book level. You can set the calculations to absorb overhead at the level of the cost organization, inventory organization, item category, or item.

**Using Expense Pools and Cost Element Groups to Calculate Overhead Absorption: Example**

This example illustrates how to use expense pools and cost element groups to define overhead accounting rules, and calculate overhead absorption for transactions.
**Scenario**

Your cost organization is a bicycle retail store with monthly overhead costs of $10,000 for rent, $500 for water, $1,500 for electricity, and $1,000 for gas. You have additional costs of $50 freight per incoming receipt; and $10 inspection fees per unit. You want to calculate overhead absorption for 5 transactions during the month, 2 receipts of bike X, and 3 receipts of bike Y.

**Transaction Details**

You could combine the water, electricity, and gas costs into one utility expense pool of $3,000. Then define the expense pools on the Manage Overhead Expense Pools page as follows.

<table>
<thead>
<tr>
<th>Expense Pool</th>
<th>Cost Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rent</td>
<td>Warehouse OH</td>
</tr>
<tr>
<td>Utilities</td>
<td>Warehouse OH</td>
</tr>
<tr>
<td>Freight</td>
<td>Freight OH</td>
</tr>
<tr>
<td>Inspection</td>
<td>Warehouse OH</td>
</tr>
</tbody>
</table>

Next define a Materials cost element group on the Manage Overhead Cost Element Groups page. This cost element group will be used to calculate overhead costs as a percentage of material costs. Your material costs are $500 per unit for bike X, and $300 per unit for bike Y.

Finally, you want overhead costs to include cost organization administrative and inventory organization facilities costs.

**Analysis**

Define the overhead accounting rules and absorption rates so that, over the course of the month, the total amount absorbed by the transactions will equal the overhead expense pools.

<table>
<thead>
<tr>
<th>Overhead Accounting Rule</th>
<th>Transaction Group</th>
<th>Transaction Type</th>
<th>Item or Item Category</th>
<th>Expense Pool</th>
<th>Cost Element</th>
<th>Cost Basis</th>
<th>Based On Cost Element Group</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule1</td>
<td>Purchase Order Transaction</td>
<td>Purchase Order Receipt</td>
<td>Item Bike X</td>
<td>Rent</td>
<td>Warehouse OH</td>
<td>Per unit</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>Rule2</td>
<td>Purchase Order Transaction</td>
<td>Purchase Order Receipt</td>
<td>Item Bike Y</td>
<td>Rent</td>
<td>Warehouse OH</td>
<td>Per transaction</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>Rule3</td>
<td>Purchase Order Transaction</td>
<td>Purchase Order Receipt</td>
<td>Category Bike</td>
<td>Rent</td>
<td>Warehouse OH</td>
<td>Percentage</td>
<td>Materials</td>
<td>45%</td>
</tr>
<tr>
<td>Rule4</td>
<td>Purchase Order Transaction</td>
<td>Purchase Order Receipt</td>
<td>Category Bike</td>
<td>Utilities</td>
<td>Warehouse OH</td>
<td>Percentage</td>
<td>Materials</td>
<td>12%</td>
</tr>
<tr>
<td>Rule5</td>
<td>Purchase Order Transaction</td>
<td>Purchase Order Receipt</td>
<td>Category Bike</td>
<td>Freight</td>
<td>Freight OH</td>
<td>Per transaction</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>
Following are the overhead cost calculations based on the rules defined.

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Item</th>
<th>Quantity</th>
<th>Rent</th>
<th>Utilities</th>
<th>Freight</th>
<th>Inspection</th>
<th>Cost Organizational Costs</th>
<th>Inventory Organization Facilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bike X</td>
<td>10</td>
<td>2,000</td>
<td>600</td>
<td>50</td>
<td>100</td>
<td>250</td>
<td>250</td>
</tr>
<tr>
<td>2</td>
<td>Bike X</td>
<td>15</td>
<td>2,750</td>
<td>900</td>
<td>50</td>
<td>150</td>
<td>375</td>
<td>375</td>
</tr>
<tr>
<td>3</td>
<td>Bike Y</td>
<td>10</td>
<td>1,350</td>
<td>360</td>
<td>50</td>
<td>100</td>
<td>150</td>
<td>150</td>
</tr>
<tr>
<td>4</td>
<td>Bike Y</td>
<td>15</td>
<td>2,025</td>
<td>540</td>
<td>50</td>
<td>150</td>
<td>225</td>
<td>225</td>
</tr>
<tr>
<td>5</td>
<td>Bike Y</td>
<td>10</td>
<td>1,350</td>
<td>360</td>
<td>50</td>
<td>100</td>
<td>150</td>
<td>150</td>
</tr>
<tr>
<td>Total</td>
<td>Overhead Absorption</td>
<td>9,475</td>
<td>2,760</td>
<td>250</td>
<td>600</td>
<td>1,150</td>
<td>1,150</td>
<td></td>
</tr>
</tbody>
</table>

**Overhead Accounting Rules: Explained**

Overhead accounting rules establish how to absorb overhead costs into inventory value and into cost of goods sold. The overheads processor checks for the rule based on the type of transaction. If a rule is defined and set to active, the processor applies overhead absorption to the transaction.

The following describes the overhead accounting rule attributes and cost drivers.

**Overhead Accounting Rule Attributes**

Associate an overhead accounting rule with a cost organization, cost book, and expense pool. The cost element from the expense pool definition is displayed automatically.

Also specify the following attributes:

- Transaction group (mandatory) and transaction type (optional). The transaction groups are predefined and they include one or more transaction types. You can define overhead rules at the transaction group level, or at the transaction type level. The transaction group options are Interorganization Transfers, Intraorganization Transfers, Inventory Transactions, Purchase Order Transactions, Sales Order Issues, and
Sales Order Returns. The transaction group controls the transaction type options, which are more granular. If the transaction type detail is not provided, then the overhead absorption occurs for all transaction types within the transaction group.

- **Transaction flow** (mandatory). Options are Issue or Receipt.
- **Inventory organization**. Required only when absorbing overhead at the level of the inventory organization. If this attribute is blank, then the overhead is applied to all transactions in all inventory organizations under the cost organization.
- **Category name and item**. Required only if you are absorbing overhead at the item category or item level.

**Cost Drivers**

In addition to the attributes, specify the cost drivers for the rule.

The cost drivers include:

- **Cost basis** (required). Options are Per Lot, Per Transaction, Per Unit, or Percentage Value. Lot is based on the standard lot size defined in the item master. The processor divides the per lot overhead rate by the standard lot size to arrive at the per unit overhead cost. For example, suppose the lot size is 100 units, the overhead rate is $10 per lot, and the quantity is 150 units; then the overhead cost per unit is $10/100 = $0.1; and the overhead absorbed is $0.1 * 150.
- **Based on**. Mandatory if the cost basis is Percentage Value, and it specifies the cost element group that the percentage is based on.
- **Rate** (mandatory). Represents either the overhead percentage amount that you want to apply to the predefined cost element group, or the currency amount that you want to apply per unit or per transaction.
- **Absorption type** (mandatory). Options are Include in Inventory, and Expense. The following are examples of different kinds of absorption:
  - Absorb to inventory value when overhead is applied to incoming transactions, including transfers from other inventory organizations.
  - Absorb and redirect as a period expense when overhead is applied to incoming transactions.
  - Absorb overhead from the expense pool and redirect to cost of goods sold when overhead is applied to outgoing transactions.

**FAQs for Define Overheads**

**Can I delete or inactivate a cost element group?**

Yes. You can inactivate a cost element group by first inactivating all rules where it is referenced. However, you cannot delete a cost element group after it has been used to define an overhead accounting rule because historical records are maintained for audit purposes.
Can I delete or inactivate an expense pool?

Yes. You can inactivate an expense pool by first inactivating all rules where it is referenced. However, you cannot delete an expense pool after it has been used to define an overhead accounting rule because historical records are maintained for audit purposes.

Can I delete or inactivate an overhead accounting rule?

Yes. You can inactivate overhead accounting rules. However you cannot delete overhead accounting rules that have been used to calculate overhead absorption in any transactions because historical records are maintained for audit purposes.

Define Subledger 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:

<table>
<thead>
<tr>
<th>Link Journal Lines Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>No link is established.</td>
</tr>
<tr>
<td>Copy from corresponding line</td>
<td>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.</td>
</tr>
<tr>
<td>Business flow</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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.

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:
The second priority will create an output from a constant value (0.9100030.50034206331.0.0.0.0). 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:

<table>
<thead>
<tr>
<th>Source</th>
<th>De-limiter</th>
<th>Segment</th>
<th>Operator</th>
<th>Value</th>
<th>De-limiter</th>
<th>Segment</th>
<th>And Or</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Distribution Account&quot;</td>
<td>.</td>
<td>&quot;Cost Center&quot;</td>
<td>=</td>
<td>&quot;Liability Account&quot;</td>
<td>.</td>
<td>&quot;Cost Center&quot;</td>
<td>'AND'</td>
</tr>
<tr>
<td>&quot;Asset Flag&quot;</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Account</th>
<th>Invoice 1</th>
<th>Invoice 2</th>
<th>Asset Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution Account</td>
<td>02-640-2210-1234</td>
<td>01-780-6120-0000</td>
<td>Yes</td>
</tr>
<tr>
<td>Liability Account</td>
<td>01-640-2210-0000</td>
<td>02-782-2210-0000</td>
<td>Yes</td>
</tr>
</tbody>
</table>

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.
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.

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.

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 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.

**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.
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.

**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).

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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounted Amount Overwrite</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td>Y - Overwrite accounted amount</td>
</tr>
<tr>
<td>Indicator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N - Not overwrite accounted amount</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounting Date</td>
<td>Date</td>
<td>Header</td>
<td>Event Class and Journal Entry Rule Set</td>
<td>Yes</td>
<td>Should be in open general ledger period</td>
</tr>
<tr>
<td>Accrual Reversal GL Date</td>
<td>Date</td>
<td>Header</td>
<td>Event Class and Journal Entry Rule Set</td>
<td>No</td>
<td>Should be later than the accounting date</td>
</tr>
</tbody>
</table>

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.
<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounting Reversal Distribution Type</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>Yes, if another accounting reversal accounting attribute is assigned.</td>
<td>Y - Reverse without creating a replacement line&lt;br&gt; B - Reverse and create a new line as replacement&lt;br&gt; N or Null - Not a reversal</td>
</tr>
<tr>
<td>Accounting Reversal First Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>Yes, if another accounting reversal accounting attribute is assigned.</td>
<td>Y - Reverse without creating a replacement line&lt;br&gt; B - Reverse and create a new line as replacement&lt;br&gt; N or Null - Not a reversal</td>
</tr>
<tr>
<td>Accounting Reversal Second Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>No</td>
<td>Y - Reversal transaction object header&lt;br&gt; N or null - Standard transaction object header</td>
</tr>
<tr>
<td>Accounting Reversal Third Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>No</td>
<td>Y - Reversal transaction object header&lt;br&gt; N or null - Standard transaction object header</td>
</tr>
<tr>
<td>Accounting Reversal Fourth Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>No</td>
<td>Y - Reversal transaction object header&lt;br&gt; N or null - Standard transaction object header</td>
</tr>
<tr>
<td>Accounting Reversal Fifth Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>No</td>
<td>Y - Reversal transaction object header&lt;br&gt; N or null - Standard transaction object header</td>
</tr>
<tr>
<td>Transaction Accounting Reversal Indicator</td>
<td>Alphanumeric</td>
<td>Header</td>
<td>Event Class</td>
<td>No</td>
<td>Y - Reversal transaction object header&lt;br&gt; N or null - Standard transaction object header</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applied to Amount</td>
<td>Number</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Applied to First System Transaction Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td></td>
</tr>
<tr>
<td>Applied to Second System Transaction Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Applied to Third System Transaction Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Applied to Fourth System Transaction Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td></td>
</tr>
<tr>
<td>Applied to Distribution Type</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td></td>
</tr>
<tr>
<td>Applied to First Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td></td>
</tr>
<tr>
<td>Applied to Second Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Applied to Third Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Applied to Fourth Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Applied to Fifth Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Applied to Application ID</td>
<td>Number</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>Yes, if another accounting attribute in the same group has assignment. Must be a valid application ID</td>
<td></td>
</tr>
<tr>
<td>Applied to Entity Code</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>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</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution Type</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>First Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Second Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subledger Document Sequence Category</td>
<td>Alphanumeric</td>
<td>Header</td>
<td>Event Class</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td></td>
</tr>
<tr>
<td>Subledger Document Sequence Identifier</td>
<td>Number</td>
<td>Header</td>
<td>Event Class</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td></td>
</tr>
<tr>
<td>Subledger Document Sequence Value</td>
<td>Number</td>
<td>Header</td>
<td>Event Class</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entered Currency Code</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>Yes</td>
<td>A valid currency code</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounted Amount</td>
<td>Number</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Conversion Date</td>
<td>Date</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Conversion Rate</td>
<td>Number</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Conversion Rate Type</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td>A valid general ledger conversion rate type or User</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail Tax Distribution Reference</td>
<td>Number</td>
<td>Line</td>
<td>Event Class</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Detail Tax Line Reference</td>
<td>Number</td>
<td>Line</td>
<td>Event Class</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Summary Tax Line Reference</td>
<td>Number</td>
<td>Line</td>
<td>Event Class</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Party Identifier</td>
<td>Number</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td>If party type C - Should be a valid customer account If party type is S - Should be a valid supplier identifier</td>
</tr>
<tr>
<td>Party Site Identifier</td>
<td>Number</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td>If party type C - Should be a valid customer account If party type is S - Should be a valid supplier identifier</td>
</tr>
<tr>
<td>Party Type</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td>C for Customer S for Supplier</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exchange Gain Account</td>
<td>Number</td>
<td>Header</td>
<td>Event Class</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Exchange Loss Account</td>
<td>Number</td>
<td>Header</td>
<td>Event Class</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gain or Loss Reference</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer to GL Indicator</td>
<td>Alphanumeric</td>
<td>Header</td>
<td>Event Class</td>
<td>No</td>
<td>Should be Y or N</td>
</tr>
</tbody>
</table>

FAQs for Define Subledger Accounting Rules

**How can I create subledger account rules and subledger journal entry rule sets for cost management?**

Create your subledger account rules on the Manage Account Rules page. It is recommended that you highlight the account rules predefined by Oracle, copy, and modify them as needed.

Create your subledger journal entry rule sets on the Manage Subledger Journal Entry Rule Sets page. It is recommended that you highlight the journal entry rule sets predefined by Oracle, copy, and modify them as needed. For each journal line rule specify the copied account combination rule.

Access both the Manage Account Rules page and Manage Subledger Journal Entry Rule Sets page from an Oracle Fusion Applications Functional Setup Manager implementation project.

Note
You must customize the predefined account rules and journal entry rule sets before proceeding with the setup of subledger accounting rules for cost management.
Define Subledger 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:

<table>
<thead>
<tr>
<th>Link Journal Lines Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>No link is established.</td>
</tr>
<tr>
<td>Copy from corresponding line</td>
<td>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.</td>
</tr>
<tr>
<td>Business flow</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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.

---

**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:
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:

<table>
<thead>
<tr>
<th>Source</th>
<th>De-limiter</th>
<th>Segment</th>
<th>Operator</th>
<th>Value</th>
<th>De-limiter</th>
<th>Segment</th>
<th>And Or</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Distribution Account&quot;</td>
<td>.</td>
<td>&quot;Cost Center&quot;</td>
<td>=</td>
<td>&quot;Liability Account&quot;</td>
<td>.</td>
<td>&quot;Cost Center&quot;</td>
<td>'AND'</td>
</tr>
<tr>
<td>&quot;Asset Flag&quot;</td>
<td>=</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Account</th>
<th>Invoice 1</th>
<th>Invoice 2</th>
<th>Asset Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution Account</td>
<td>02-640-2210-1234</td>
<td>01-780-6120-0000</td>
<td>Yes</td>
</tr>
<tr>
<td>Liability Account</td>
<td>01-640-2210-0000</td>
<td>02-782-2210-0000</td>
<td>Yes</td>
</tr>
</tbody>
</table>

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.
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.

---

**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.

---

**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 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.

**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:

![Accounting Event Model for Loans](image-url)
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 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.
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.

**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).

### 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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounted Amount Overwrite Indicator</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td>Y - Overwrite accounted amount</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N - Not overwrite accounted amount</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounting Date</td>
<td>Date</td>
<td>Header</td>
<td>Event Class and Journal Entry Rule Set</td>
<td>Yes</td>
<td>Should be in open general ledger period</td>
</tr>
<tr>
<td>Accrual Reversal GL Date</td>
<td>Date</td>
<td>Header</td>
<td>Event Class and Journal Entry Rule Set</td>
<td>No</td>
<td>Should be later than the accounting date</td>
</tr>
</tbody>
</table>

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.
<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounting Reversal Distribution Type</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>Yes, if another accounting reversal accounting attribute is assigned.</td>
<td></td>
</tr>
<tr>
<td>Accounting Reversal First Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>Yes, if another accounting reversal accounting attribute is assigned.</td>
<td></td>
</tr>
<tr>
<td>Accounting Reversal Second Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Accounting Reversal Fourth Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Accounting Reversal Fifth Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Accounting Reversal Indicator</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>Yes, if another accounting reversal accounting attribute is assigned.</td>
<td>Y - Reverse without creating a replacement line</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B - Reverse and create a new line as replacement</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N or Null - Not a reversal</td>
</tr>
<tr>
<td>Transaction Accounting Reversal Indicator</td>
<td>Alphanumeric</td>
<td>Header</td>
<td>Event Class</td>
<td>No</td>
<td>Y - Reversal transaction object header</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N or null - Standard transaction object header</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applied to Amount</td>
<td>Number</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Applied to First System Transaction Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td></td>
</tr>
<tr>
<td>Applied to Second System Transaction Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Applied to Third System Transaction Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td></td>
</tr>
<tr>
<td>Applied to Fourth System Transaction Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Applied to Distribution Type</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td></td>
</tr>
<tr>
<td>Applied to First Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td></td>
</tr>
<tr>
<td>Distribution Identifier</td>
<td>Data Type</td>
<td>Journal Entry Level</td>
<td>Assignment to Rules</td>
<td>Assignment Required?</td>
<td>Validation Rules</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------</td>
<td>---------------------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>First Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Second Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Distribution Type</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Applied to Second Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Applied to Third Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Applied to Fourth Distribution Identifier</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Applied to Fifth Distribution Identifier</td>
<td>Number</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td>Must be a valid application ID</td>
</tr>
<tr>
<td>Applied to Entity Code</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td>Must be a valid Entity for the application selected in Applied to Application ID</td>
</tr>
</tbody>
</table>

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.
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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subledger Document Sequence Category</td>
<td>Alphanumeric</td>
<td>Header</td>
<td>Event Class</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td></td>
</tr>
<tr>
<td>Subledger Document Sequence Identifier</td>
<td>Number</td>
<td>Header</td>
<td>Event Class</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td></td>
</tr>
<tr>
<td>Subledger Document Sequence Value</td>
<td>Number</td>
<td>Header</td>
<td>Event Class</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entered Currency Code</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>Yes</td>
<td>A valid currency code</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounted Amount</td>
<td>Number</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Conversion Date</td>
<td>Date</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Conversion Rate</td>
<td>Number</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Conversion Rate Type</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>No</td>
<td>A valid general ledger conversion rate type or User</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail Tax Distribution Reference</td>
<td>Number</td>
<td>Line</td>
<td>Event Class</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Detail Tax Line Reference</td>
<td>Number</td>
<td>Line</td>
<td>Event Class</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Summary Tax Line Reference</td>
<td>Number</td>
<td>Line</td>
<td>Event Class</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Party Identifier</td>
<td>Number</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
</tr>
<tr>
<td>Party Site Identifier</td>
<td>Number</td>
<td>Line</td>
<td>Event Class and Journal Line Rule</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
</tr>
<tr>
<td>Party Type</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>Yes, if another accounting attribute in the same group has assignment.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exchange Gain Account</td>
<td>Number</td>
<td>Header</td>
<td>Event Class</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Exchange Loss Account</td>
<td>Number</td>
<td>Header</td>
<td>Event Class</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gain or Loss Reference</td>
<td>Alphanumeric</td>
<td>Line</td>
<td>Event Class</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Accounting Attributes</th>
<th>Data Type</th>
<th>Journal Entry Level</th>
<th>Assignment to Rules</th>
<th>Assignment Required?</th>
<th>Validation Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer to GL Indicator</td>
<td>Alphanumeric</td>
<td>Header</td>
<td>Event Class</td>
<td>No</td>
<td>Should be Y or N</td>
</tr>
</tbody>
</table>

FAQs for Define Subledger Accounting Rules

How can I create subledger account rules and subledger journal entry rule sets for receipt accounting?

Create your subledger account rules on the Manage Account Rules page. It is recommended that you highlight the account rules predefined by Oracle, copy, and modify them as needed.

Create your subledger journal entry rule sets on the Manage Subledger Journal Entry Rule Sets page. It is recommended that you highlight the journal entry rule sets predefined by Oracle, copy, and modify them as needed. For each journal line rule specify the copied account combination rule.

Access both the Manage Account Rules page and Manage Subledger Journal Entry Rule Sets page from an Oracle Fusion Applications Functional Setup Manager implementation project.

Note
You must customize the predefined account rules and journal entry rule sets before proceeding with the setup of subledger accounting rules for receipt accounting.
Define Transactional Business Intelligence Configuration: Highlights

Configure Oracle Transactional Business Intelligence for ad hoc reporting, review certain setup objects to be used in Transactional Business Intelligence, and manage the presentation catalog and currency type display.

**Defining Transactional Business Intelligence Configuration**
- Review details about the Transactional Business Intelligence tasks. Refer to the Oracle Fusion Transactional Business Intelligence Administrator's Guide.

**Access to Person Data**

**Assigning Security Profiles to Job Roles for Oracle Fusion Transactional Business Intelligence Users: Explained**

Users of Oracle Fusion Transactional Business Intelligence (Transactional Business Intelligence) need access to some person data for reporting purposes. To provide this access, you assign a predefined security profile to relevant job or abstract roles using the Oracle Fusion Human Capital Management (HCM) setup task Manage Data Role and Security Profiles. On completion of this task, Oracle Fusion Data Security is updated automatically for roles being used to access Transactional Business Intelligence.

**Job or Abstract Roles and Related Security Profiles**

The following table identifies, by Oracle Fusion product, the job and abstract roles that need access to person data and the predefined security profile that you assign to each role.

<table>
<thead>
<tr>
<th>Product</th>
<th>Job or Abstract Role</th>
<th>Security Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Fusion Assets</td>
<td>Asset Accountant</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Application</td>
<td>Role</td>
<td>Security Profile</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>-------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Oracle Fusion Assets</td>
<td>Asset Accounting Manager</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Enterprise Planning and Budgeting</td>
<td>Budget Analyst</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Enterprise Planning and Budgeting</td>
<td>Budget Manager</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Financial Consolidation Hub</td>
<td>Consolidation Accountant</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Financial Consolidation Hub</td>
<td>Consolidation Manager</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Financials Common Module</td>
<td>Intercompany Accountant</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion General Ledger</td>
<td>Financial Analyst</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion General Ledger</td>
<td>General Accountant</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion General Ledger</td>
<td>General Accounting Manager</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Incentive Compensation</td>
<td>Incentive Compensation Participant Manager</td>
<td>View Manager Hierarchy</td>
</tr>
<tr>
<td>Oracle Fusion Inventory Management</td>
<td>Warehouse Manager</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Project Foundation</td>
<td>Project Accountant</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Project Foundation</td>
<td>Project Administrator</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Project Foundation</td>
<td>Project Billing Specialist</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Purchasing</td>
<td>Buyer</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Sourcing</td>
<td>Category Manager</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Sourcing</td>
<td>Sourcing Project Collaborator</td>
<td>View All Workers</td>
</tr>
</tbody>
</table>

For example, as part of their Transactional Business Intelligence setup:

- Oracle Fusion Assets implementors must assign the predefined security profile View All Workers to the Asset Accountant and Asset Accounting Manager job roles.

- Oracle Fusion Incentive Compensation implementors must assign the predefined security profile View Manager Hierarchy to the abstract role Incentive Compensation Participant Manager.

The security profiles that HCM roles need to access Transactional Business Intelligence are assigned during the setup of HCM data security: no additional setup is required for Transactional Business Intelligence purposes.

**Enabling an Oracle Fusion Transactional Business Intelligence User to Access Person Data: Worked Example**

This example shows how to assign a security profile to a job or abstract role to enable users with that role to access person data. This task is required for users of Oracle Fusion Transactional Business Intelligence (Transactional Business Intelligence) who do not also use Oracle Fusion Human Capital Management (HCM).
The following table summarizes key decisions for this scenario. When performing this task, use the job or abstract role for your product and the name of the relevant predefined person security profile in place of those shown here.

<table>
<thead>
<tr>
<th>Decisions to Consider</th>
<th>In This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the name of the Transactional Business Intelligence job or abstract role?</td>
<td>Warehouse Manager</td>
</tr>
<tr>
<td>What is the name of the person security profile?</td>
<td>View All Workers</td>
</tr>
</tbody>
</table>

To perform these tasks, you must have the role IT Security Manager.

1. Launch the task Manage Data Role and Security Profiles.
2. Search for the job or abstract role.
3. Assign the relevant predefined security profile to the job or abstract role.

**Launching the Task Manage Data Role and Security Profiles**

1. On the Overview page of the Setup and Maintenance work area, click the **All Tasks** tab.
2. In the Search region, complete the fields as shown in this table:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search</td>
<td>Tasks</td>
</tr>
<tr>
<td>Name</td>
<td>Manage Data Role and Security Profiles</td>
</tr>
</tbody>
</table>

3. Click **Search**.
4. In the search results, click **Go to Task** for the Manage Data Role and Security Profiles task.

**Searching for the Job or Abstract Role**

1. On the Manage HCM Data Roles page, enter the job name Warehouse Manager in the **Role** field.
2. Click **Search**.
3. In the search results, highlight the entry for the Warehouse Manager job role.
4. Click **Assign**.

**Assigning the Security Profile to the Job Role**

1. In the Person Security Profile field on the Assign Data Role: Security Criteria page, select the security profile View All Workers.
2. Click **Review**.
3. On the Assign Data Role: Review page, click **Submit**.
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
Customizing Pages Using Oracle Composer: Highlights

You can customize dashboards and some work areas, where available, for all or some users based on a selected customization layer, for example only for users in a specific country or with a specific job role. When you select to customize a page from the Administration menu in the global area, you invoke Oracle Composer, which enables the customization. The Administration menu is only available if you have appropriate roles.

From the same menu, you can also:

- Customize the global area at the site layer.
- Access the Customization Manager, which displays a list of components in the current page and details about the layers in which each component is customized.
- Access sandboxes from the Administration menu, to make customizations to a runtime use session before deploying your changes to the mainline.

Customizing pages using Oracle Composer, the Customization Manager, and sandboxes are described in the Oracle Fusion Applications Extensibility Guide.

Editing Pages

- Customizations that you can make to existing pages include adding content and changing layout.

  See: Editing a Page in Oracle Composer

- You can also update display and other options for specific components within the page.

  See: Editing Component Properties in Oracle Composer

Customization Manager

- Use Customization Manager to analyze and diagnose customization metadata, and to perform customization related tasks that cannot be done in the user interface, for example to revert customizations to a previous version. You can also do direct customization by manipulating the metadata and uploading it back.
See: Using Customization Manager to Manage Runtime Customizations

**Sandbox**

- Create or select an appropriate sandbox, and set it as active to capture your customizations using Oracle Composer. When you are ready, you publish the sandbox to make your changes available to users.

See: Using the Sandbox Manager

**Sandboxes: Highlights**

Use a sandbox to commit customizations to a runtime use session for validation before deploying changes to the mainline. Administrators create and manage sandboxes. An active sandbox isolates changes from the mainline and other users.

Sandboxes can contain the following types of customization changes.

- Metadata, such as non-flexfield UI page customizations
- Data security
- Generated flexfields business components

Metadata changes are captured in a metadata sandbox. Data security changes are additionally captured in a data security enabled sandbox. Changes to a flexfield are captured in a flexfield that is deployed as a single flexfield sandbox. Once you are ready to make sandbox changes available in the mainline, you either publish the metadata or data security sandbox, or deploy the flexfield. Only metadata and data security sandboxes can be downloaded as a sandbox file for import to another Oracle Fusion Applications instance.

The following table lists the differences among the types of sandboxes.

<table>
<thead>
<tr>
<th>Type of Changes</th>
<th>Type of Sandbox</th>
<th>Method for Making Changes Available in Mainline</th>
<th>Downloadable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metadata</td>
<td>Sandbox</td>
<td>Publish sandbox</td>
<td>Yes</td>
</tr>
<tr>
<td>Data security</td>
<td>Sandbox enabled for data security changes</td>
<td>Publish sandbox</td>
<td>Yes</td>
</tr>
<tr>
<td>Flexfield</td>
<td>Flexfield deployed as a flexfield-enabled sandbox</td>
<td>Deploy flexfield</td>
<td>No</td>
</tr>
</tbody>
</table>

Only one sandbox can be active at a time. Changes made while a sandbox is active are captured in that sandbox.

For more information on using the Sandbox Manager, and customizing and securing pages, business objects, data, and custom objects in a sandbox, see the Oracle Fusion Applications Extensibility Guide.
Managing a Page Customization Sandbox

You can make metadata (MDS) type changes in a sandbox, including menu customizations, changes to the personalization menu, implicit ADF customizations, or changes made with Oracle Composer or CRM Application Composer.

- If you are entitled to do so, manage sandboxes in the Sandbox Manager.

  See: Using the Sandbox Manager

- Implement customizations on an existing page to change the metadata of a sandbox before deploying the changes to the mainline.

  See: Customizing Existing Pages

- Using CRM Application Composer, customize business objects in a sandbox before deploying the changes to the mainline.

  See: Customizing Objects

Managing a Data Security Sandbox

You can create a sandbox for data security testing, or designate an existing sandbox to become enabled for data security testing.

- If you are entitled to do so, manage data security-enabled sandboxes in the Sandbox Manager.

  See: Using the Sandbox Manager

- If you customize business objects in CRM Application Composer, you may need to define data security policies to make them accessible to users.

  See: Defining Security Policies for Business Objects

- If you create new business objects, you need to secure them.

  See: Customizing Security for Custom Business Objects

Managing a Flexfield Sandbox

You create a flexfield-enabled sandbox by deploying one flexfield to a sandbox using the Manage Flexfield task flow. The flexfield sandbox gets its name from the flexfield you deploy. You cannot test two flexfields in the same sandbox. Once you deploy a flexfield as a sandbox, you must sign out and back in to view how the sandbox runtime reflects the flexfield changes, such as new segments. You can redeploy the same flexfield to the same sandbox repeatedly as you make incremental changes to the flexfield setup.

- Since a flexfield sandbox cannot be published, any page customizations or data security in the flexfield sandbox cannot reach the mainline when the flexfield is deployed to the mainline. If you have entitlement to do so, see Deploying a Flexfield to a Sandbox: Points to Consider.

- If you are entitled to do so, manage flexfield-enabled sandboxes in the Sandbox Manager.
See: Using the Sandbox Manager
Importing and Exporting Setup Data

Configuration Packages: Explained

Almost all Oracle Fusion application implementations require moving functional setup data from one instance into another at various points in the lifecycle of the applications. For example, one of the typical cases in any enterprise application implementation is to first implement in a development or test application instance and then deploy to a production application instance after thorough testing. You can move functional setup configurations of applications from one application instance into another by exporting and importing Configuration packages from the Manage Configuration Packages page.

A Configuration Package contains the setup import and export definition. The setup import and export definition is the list of setup tasks and their associated business objects that identifies the setup data for export as well as the data itself. When you create a configuration package only the setup export and import definition exists. Once you export the configuration package appropriate setup data is added to the configuration package using the definition. Once a configuration package is exported, the setup export and import definition is locked and cannot be changed.

You generate the setup export and import definition by selecting an implementation project and creating a configuration package. The tasks and their associated business objects in the selected implementation project define the setup export and import definition for the configuration package. In addition, the sequence of the tasks in the implementation project determine the export and import sequence.

Exporting and Importing Setup Data: Explained

A configuration package is required to export setup data. You can export a configuration package once you create it, or at any time in the future. During export, appropriate setup data will be identified based on the setup export definition and added to the configuration package. The setup data in the configuration package is a snapshot of the data in the source application instance at the time of export. After the export completes, you can download the configuration package as a zipped archive of multiple XML files, move it to the target application instance, and upload and import it.

Export

You can export a configuration package multiple times by creating multiple versions. While the export definition remains the same in each version, the setup data can be different if you modified the data in the time period between the different runs of the export process. Since each version of the configuration package has a snapshot of the data in the source instance, you can compare and
analyze various versions of the configuration package to see how the setup data changed.

**Import**

In the target application instance, the setup import process will insert all new data from the source configuration package that does not already exist and update any existing data with changes from the source. Setup data that exists in the target instance but not in source will remain unchanged.

**Export and Import Reports**

You can review the results of the export and import processes using reports. The results appear ordered by business objects and include information on any errors encountered during the export or import process. If a setup export or import process paused due to errors encountered or for a manual task to be performed outside of the application, then you can resume the paused process. These reports show what setup data was exported or imported and by which specific process. You can change the reports to validate the setup data as well as to compare or analyze it. A report is generated for each business object. These reports show the same information as the export and import results seen directly in the application.

Process status details are available as text files showing the status of an export or import process including the errors encountered during the process.

---

**Moving Common Reference Objects**

**Moving Common Reference Objects: Overview**

The common reference objects in Oracle Fusion Middleware Extensions for Applications are used by several setup tasks in the Setup and Maintenance work area. The common reference objects become a part of the configuration package that is created for an implementation project. While moving the application content, for example, from the test phase to the production phase of an implementation, you must pay special attention to the nuances of these common reference objects.

**Parameters**

The common reference objects are represented as business objects. A single object can be referenced in multiple setup tasks with different parameters. In the configuration package that is created for the implementation project, parameters passed to a setup task are also passed to the business objects being moved. As a result, the scope of the setup tasks is maintained intact during the movement.

**Dependencies**

Common reference objects may have internal references or dependencies among other common reference objects. Therefore, it is necessary that all the dependencies are noted before the movement of objects so that there are no broken references among the objects.

**Business Objects for Moving Common Reference Objects: Points to Consider**

Common reference objects in Oracle Fusion Functional Setup Manager are represented by business objects. These business objects are the agents that contain the application content and carry them across whenever the application setup is moved from one environment to another, for example, test environment to production environment.

**Choice of Parameters**

The following table lists the business objects, the corresponding movement details, and the effect of the setup task parameter on the scope of the movement.
Only the translation in the current user language is moved.

<table>
<thead>
<tr>
<th>Business Object Name</th>
<th>Moved Functional Item</th>
<th>Effect on the Scope of Movement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Menu Customization</td>
<td>Customizations to the navigator menu</td>
<td>Movement of navigator menu customizations requires manual effort. For details regarding the manual command, refer to the Oracle Fusion Applications Administrators Guide.</td>
</tr>
<tr>
<td>Application Message</td>
<td>Messages and associated tokens</td>
<td>No parameters: all messages are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>moduleType/moduleKey</code>: only messages belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>messageName/applicationId</code>: only the specified message is moved.</td>
</tr>
<tr>
<td>Application Taxonomy</td>
<td>Application taxonomy modules and components</td>
<td>No parameters: all taxonomy modules and components are moved.</td>
</tr>
<tr>
<td>Application Attachment Entity</td>
<td>Attachment entities</td>
<td>No parameters: all attachment entities are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>moduleType/moduleKey</code>: only attachment entities belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</td>
</tr>
<tr>
<td>Application Attachment Category</td>
<td>Attachment categories and category-to-entity mappings</td>
<td>No parameters: all attachment categories and category-to-entity mappings are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>moduleType/moduleKey</code>: only attachment categories belonging to the specified module and its descendant modules in the taxonomy hierarchy along with the respective category-to-entity mappings are moved.</td>
</tr>
<tr>
<td>Application Document Sequence Category</td>
<td>Document sequence categories</td>
<td>No parameters: all categories are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>moduleType/moduleKey</code>: only categories belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>code/applicationId</code>: only the specified document sequence category code is moved.</td>
</tr>
<tr>
<td>Application Document Sequence</td>
<td>Document sequences and their assignments</td>
<td>No parameters: all sequences are moved. moduleType/moduleKey: only document sequences belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved name: only the specified document sequence is moved.</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Application Descriptive Flexfield</td>
<td>Descriptive flexfield registration data and setup data</td>
<td>No parameters: all descriptive flexfields are moved. moduleType/moduleKey: only descriptive flexfields belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved. descriptiveFlexfieldCode/applicationId: only the specified descriptive flexfield is moved.</td>
</tr>
</tbody>
</table>

**Note**

Importing a flexfield’s metadata can change its deployment status and therefore, the affected flexfields must be redeployed. The import process automatically submits affected flexfields for redeployment.

**Note**

Only flexfields with a deployment status of Deployed or Deployed to Sandbox are eligible to be moved.
| Application Extensible Flexfield | Extensible flexfield registration data and setup data, including categories | No parameters: all extensible flexfields are moved
moduleType/moduleKey: only extensible flexfields belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.

extensibleFlexfieldCode/applicationId: only the specified extensible flexfield is moved.

---

| Application Key Flexfield | Key flexfield registration data and setup data | No parameters: all key flexfields are moved.
moduleType/moduleKey: only key flexfields belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.

keyFlexfieldCode/applicationId: only the specified key flexfield is moved.

---

| Application Extensible Flexfield | Extensible flexfield registration data and setup data, including categories | No parameters: all extensible flexfields are moved
moduleType/moduleKey: only extensible flexfields belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.

extensibleFlexfieldCode/applicationId: only the specified extensible flexfield is moved.

---

| Application Key Flexfield | Key flexfield registration data and setup data | No parameters: all key flexfields are moved.
moduleType/moduleKey: only key flexfields belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.

keyFlexfieldCode/applicationId: only the specified key flexfield is moved.

---

Note
Importing a flexfield’s metadata can change its deployment status and therefore, the affected flexfields must be redeployed. The import process automatically submits affected flexfields for redeployment.

Note
Only flexfields with a deployment status of Deployed or Deployed to Sandbox are eligible to be moved.
| Application Flexfield Value Set | Value set setup data | No parameters: all value sets are moved.  
| moduleType/moduleKey: only value sets belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.  
| valueSetCode: only the specified value set is moved.  
| Application Reference Currency | Currency data | No parameters: all currencies are moved.  
| Application Reference ISO Language | ISO language data | No parameters: all ISO languages are moved.  
| Application Reference Industry | Industry data including industries in territories data | No parameters: all industries are moved.  
| Application Reference Language | Language data | No parameters: all languages are moved.  
| Application Reference Natural Language | Natural language data | No parameters: all natural languages are moved.  
| Application Reference Territory | Territory data | No parameters: all territories are moved.  
| Application Reference Time zone | Time zone data | No parameters: all time zones are moved.  
| Application Standard Lookup | Standard lookup types and their lookup codes | No parameters: all standard lookups are moved.  
| moduleType/moduleKey: only standard lookups belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.  
| lookupType: only the specified common lookup is moved.  

<table>
<thead>
<tr>
<th>Application</th>
<th>Lookup Types and Their Lookup Codes</th>
<th>Parameters</th>
</tr>
</thead>
</table>
| Application Common Lookup | Common lookup types and their lookup codes | No parameters: all common lookups are moved.

- **moduleType/moduleKey**: only common lookups belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.
- **lookupType**: only the specified common lookup is moved.

<table>
<thead>
<tr>
<th>Application Set-Enabled Lookup</th>
<th>Set-enabled lookup types and their lookup codes</th>
<th>Parameters</th>
</tr>
</thead>
</table>
|                               |                                 | No parameters: all set-enabled lookups are moved.

- **moduleType/moduleKey**: only set-enabled lookups belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.
- **lookupType**: only the specified set-enabled lookup is moved.

<table>
<thead>
<tr>
<th>Application Profile Category</th>
<th>Profile categories</th>
<th>Parameters</th>
</tr>
</thead>
</table>
|                              |                    | No parameters: all profile categories are moved.

- **moduleType/moduleKey**: only categories belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.
- **name/applicationId**: only the specified category is moved.

<table>
<thead>
<tr>
<th>Application Profile Option</th>
<th>Profile options and their values</th>
<th>Parameters</th>
</tr>
</thead>
</table>
|                            |                                 | No parameters: all profile options and their values are moved.

- **moduleType/moduleKey**: only profile options and their values belonging to the specified module are moved.
- **profileOptionName**: only the specified profile option and its values are moved.
| Application Profile Value | Profile options and their values | No parameters: all profiles and their values are moved.  
|                          |                                 |   
|                          | ModuleType/ModuleKey: only profiles and their values belonging to the specified module are moved.  
|                          | CategoryName/CategoryApplicationId: only profiles and their values belonging to the specified category are moved.  
|                          | ProfileOptionName: only the specified profile and its values are moved.  
| Application Reference Data Set | Reference data sets | No parameters: all sets are moved.  
| Application Reference Data Set Assignment | Reference data set assignments | DeterminantType: only assignments for the specified determinant type are moved.  
|                          | DeterminantType/ReferenceGroupName: only assignments for the specified determinant type and reference group are moved.  
| Application Tree Structure | Tree structures and any labels assigned to the tree structure | No parameters: all tree structures (and their labels) are moved.  
|                          | ModuleType/ModuleKey: only tree structures (and their labels) belonging to the specified module are moved.  
|                          | TreeStructureCode: only the specified tree structure (with its labels) is moved.  
| Application Tree | Tree codes and versions | No parameters: all trees are moved.  
|                          | ModuleType/ModuleKey: only trees belonging to the specified module are moved.  
|                          | TreeStructureCode: only trees belonging to the specified tree structure are moved.  
|                          | TreeStructureCode/TreeCode: only trees belonging to the specified tree structure and tree code are moved.  

| Application Tree Label | Tree structures and any labels assigned to the tree structure | No parameters: all tree structures (and their labels) are moved.  
*moduleType/moduleKey*: only tree structures (and their labels) belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.  
*treeStructureCode*: only the specified tree structure (with its labels) is moved. |
|------------------------|-------------------------------------------------|-------------------------------------------------|
| Application Data Security Policy | Database resources, actions, conditions, and data security policies | No parameters: all database resources/actions/conditions/policies are moved.  
*moduleType/moduleKey*: only database resources/actions/conditions/policies belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.  
*objName*: only the specified database resource along with its actions/conditions/policies is moved. |

**Note**
- If the policies being moved contain reference to newly created roles, move the roles before moving the policies.
- If the source and target systems use different LDAPs, manually perform the GUID reconciliation after moving the data security policies.

---

**Moving Related Common Reference Objects: Points to Consider**

Certain common reference objects may use other common reference objects creating dependencies among the objects. During the movement of common reference objects, these dependencies or references need to be taken care of.

**Dependencies**

The dependencies among the common reference objects may be caused by any of the following conditions.

- Flexfield segments use value sets
• Value sets may make use of standard, common, or set-enabled lookups

• Key flexfields may have an associated tree structure and key flexfield segments may have an associated tree code

• Tree codes and versions may be defined over values of a value set

• Data security policies may be defined for value sets that have been enabled for data security

You may choose to move one, some, or all of the business objects by including the ones you want to move in your configuration package. For example, you may choose to move only value sets and not lookups, or you may choose to move both value sets and their lookups as part of the same package. Whatever be the combination, it is recommended that during the movement of objects, you follow an order that maintains the dependencies among the objects.

While moving the business objects, adhere to the guidelines and exactly follow the order as listed below.

1. Move created taxonomy modules before moving any objects that reference them, such as flexfields, lookups, profiles, attachments, reference data sets, document sequences, messages, and data security.

2. Move created currencies before moving any objects that reference them, such as territories.

3. Move created territories before moving any objects that reference them, such as languages and natural languages.

4. Move created ISO languages before moving any objects that reference them, such as languages, natural languages, and industries.

5. Move created tree structures before moving any objects that reference them, such as trees or tree labels.

6. Move created profile options before moving any objects that reference them, such as profile categories or profile values.

7. Move created attachment entities before moving any objects that reference them, such as attachment categories that reference them.

Note

In scenarios where there may be dependencies on other objects, you must move the dependencies before moving the referencing object. For example, if data security policies being moved have dependencies on newly created security roles, you must move the security roles before moving the security policies.

Using Seed Data Framework to Move Common Reference Objects: Points to Consider

To move the common reference objects, you can use the Seed Data Framework (SDF). You can also use the command line interface of SDF to move the object setup data. For more information about seed data loaders including common reference object loaders, see Oracle Fusion Applications Developer’s Guide.
Movement Dependencies

The seed data interface moves only the setup metadata. For example, if you use SDF to import flexfield metadata, the flexfield setup metadata is imported into your database. However, you must invoke the flexfield deployment process separately after seed data import to regenerate the runtime flexfield artifacts in the target environment. Similarly, if you use SDF to import data security metadata, you must first move any new referenced roles and then manually run the GUID reconciliation where required.

To ensure that the reference data is not lost during the movement, certain guidelines are prescribed. It is recommended that you perform the movement of object data exactly in the order given below.

Note

Only the translation in the current user language is moved.

1. Move created taxonomy modules before moving any objects that reference them, such as flexfields, lookups, profiles, attachments, reference data sets, document sequences, messages, and data security.
2. Move created currencies before moving any objects that reference them, such as territories.
3. Move created territories before moving any objects that reference them, such as languages and natural languages.
4. Move created ISO languages before moving any objects that reference them, such as languages, natural languages, and industries.
5. Move created tree structures before moving any objects that reference them, such as trees or tree labels.
6. Move created profile options before moving any objects that reference them, such as profile categories or profile values.
7. Move created attachment entities before moving any objects that reference them, such as attachment categories that reference them.
8. Move created reference data sets before moving any objects that reference them, such as reference data set assignments and set-enabled lookups.
9. Move created document sequence categories before moving any objects that reference them, such as document sequences.
10. Move created tree labels before moving any objects that reference them, such as trees.
11. Move created data security objects and policies before moving any objects that reference them, such as value sets.
12. Move created value sets before moving any objects that reference them, such as flexfields.
13. Move created trees before moving any objects that reference them, such as key flexfields.
abstract role
A description of a person’s function in the enterprise that is unrelated to the person’s job (position), such as employee, contingent worker, or line manager. A type of enterprise role.

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 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 flexfield
The chart of accounts that determines the structure, such as the number and order of individual segments, as well as the corresponding values per segment.

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.

action
The kind of access named in a security policy, such as view or edit.

actual cost
A cost method that tracks the actual cost of each receipt into inventory. When depleting inventory, the processor logically identifies the receipts that are consumed to satisfy the depletion, and assigns the associated receipt costs to the depletion.

ADF
Acronym for Application Developer Framework. A set of programming principles and rules for developing software applications.
**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.

**application role**

A role specific to applications and stored in the policy store.

**Applications Core**

Abbreviation for Oracle Fusion Middleware Extensions for Applications. The technical product code is FND.

**assignment**

A set of information, including job, position, pay, compensation, managers, working hours, and work location, that defines a worker's or nonworker's role in a legal employer.

**balancing segment**

A chart of accounts segment used to automatically balance all journal entries for each value of this segment.

**beneficiary**

A person or organization designated to receive benefits from a compensation plan on the death of the plan participant.

**BPEL**

Business Process Execution Language; a standard language for defining how to send XML messages to remote services, manipulate XML data structures, receive XML messages asynchronously from remote services, manage events and exceptions, define parallel sequences of execution, and undo parts of processes when exceptions occur.

**business function**

A business process, or an activity that can be performed by people working within a business unit and describes how a business unit is used.

**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.

calendar event
A period that signifies an event, such as a public holiday or a training course, that impacts worker availability.

chart of accounts
The account structure your organization uses to record transactions and maintain account balances.

clause adoption
Reusing a clause from the global business unit in local business units either by adopting the clause without change or by localizing it.

clause localization
A type of clause adoption where the adopted clause is edited to suit the local business unit needs.

condition
An XML filter or SQL predicate WHERE clause in a data security policy that specifies what portions of a database resource are secured.

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.

contingent worker
A self-employed or agency-supplied worker. Contingent worker work relationships with legal employers are typically of a specified duration. Any
person who has a contingent worker work relationship with a legal employer is a contingent worker.

contract deviations

Differences between the contract terms in a contract and those in the contract terms template applied to that contract and any deviations from company policies as determined by Contract Expert feature rules.

Contract Expert

A feature of the application that permits you to create business rules in the Contract Terms Library to enforce corporate policies and standards for contracts.

Contract Terms Library

A repository of standard clauses, contract terms templates, and business rules built using Contract Expert.

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 book

A view or method of cost accounting for inventory transactions. You can create multiple cost books and assign them to a cost organization for different financial and management reporting purposes.

cost center

A unit of activity or group of employees used to assign costs for accounting purposes.

cost component

Incoming costs from external sources are represented by cost components which are mapped to cost elements. Examples of cost components are purchase order material and tax.

cost component group

Mapping of cost components to cost elements. Cost component groups use set-level definitions, and are one of the attributes of item cost profiles.

cost element

The user-defined level where costs of an item are tracked through the inventory accounting life cycle; for example, the material, overhead, and tax costs of an item can be tracked as separate cost elements.
cost element group
A grouping of cost elements that is used as the basis for calculation of overhead costs, such as materials cost element group, or freight cost element group.

cost organization
A grouping of inventory organizations that indicates legal and financial ownership of inventory, and which establishes common costing and accounting policies.

cost organization book
Designates which cost book a cost organization uses for different costing and reporting purposes. For example, the Canada cost organization may use a perpetual average cost book and a primary cost book. In this case, there are two cost organization books: Canada-Perpetual Average, and Canada-Primary.

cost profile
Defines the cost accounting policies for items, such as the cost method and valuation structure.

country holding company
A legal entity that acts on behalf of several divisions within an enterprise, and is the legal employer in a country.

data dimension
A stripe of data accessed by a data role, such as the data controlled by a business unit.

data instance set
The set of human capital management (HCM) data, such as one or more persons, organizations, or payrolls, identified by an HCM security profile.

data role
A role for a defined set of data describing the job a user does within that defined set of data. A data role inherits job or abstract roles and grants entitlement to access data within a specific dimension of data based on data security policies. A type of enterprise role.

data role template
A template used to generate data roles by specifying which base roles to combine with which dimension values for a set of data security policies.

data security
The control of access to data. Data security controls what action a user can taken against which data.
data security policy
A grant of entitlement to a role on an object or attribute group for a given condition.

database resource
An applications data object at the instance, instance set, or global level, which is secured by data security policies.

default cost profile
Simplifies the effort of assigning cost profiles to items. Items can automatically inherit the default cost profile, if there is no manually-defined cost profile.

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.

descriptive flexfield
An extendable field that captures additional information.

determinant
A value that determines which reference data set will be used in a specific business context.

determinant type
Designates the field within transactional columns that controls how data is shared across organizations such as business unit, asset book, cost organization or project unit. The type determines the reference data sets that would be used in a transaction.

determinant type
An additional and optional field within transactional columns (besides category and application) that is used to assign document sequences. The available determinant types are Business Unit, Ledger, Legal Entity, and Tax Registration.
**determinant value**

A value specific to the determinant type dimension of a document sequence. The determinant value is relevant in a document sequence assignment only if the document sequence has a determinant type. If Ledger is the determinant type for a document sequence, the determinant value is the specific ledger number whose documents are numbered by the document sequence.

**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.

**duty role**

A group of function and data privileges representing one duty of a job. Duty roles are specific to applications, stored in the policy store, and shared within an Oracle Fusion Applications instance.

**enterprise**

An organization with one or more legal entities under common control.

**enterprise role**

Abstract, job, and data roles are shared across the enterprise. An enterprise role is an LDAP group. An enterprise role is propagated and synchronized across Oracle Fusion Middleware, where it is considered to be an external role or role not specifically defined within applications.

**entitlement**

Grants of access to functions and data. Oracle Fusion Middleware term for privilege.

**ESS**

Acronym for Enterprise Storage Server. An application that optimizes data storage.

**expense pool**

A grouping of similar costs representing one or more general ledger expense accounts, such as warehouse or freight expense pools. Expense pools are used in the definition of overhead accounting rules.

**extensible flexfield**

Customizable expansion space, as with descriptive flexfields, but able to capture multiple sets of information within a context and multiple contexts grouped to
appear in a named region of a user interface page. Some extensible flexfields allow grouping contexts into categories.

**feature choice**
A selection you make when configuring offerings that modifies a setup task list, or a setup page, or both.

**FIFO**
Abbreviation for first in, first out. A material control technique of rotating inventory stock so that the earliest inventory units received or produced are the first units used or shipped. The ending inventory therefore consists of the most recently acquired goods.

**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.

**function security**
The control of access to a page or a specific widget or functionality within a page. Function security controls what a user can do.

**global area**
The region across the top of the user interface. It provides access to features and tools that are relevant to any page you are on.

**grade**
A component of the employment model that defines the level of compensation for a worker.

**GTIN**
Abbreviation for Global Trade Identification Number

**HCM**
Abbreviation for Human Capital Management.
HCM data role

A job role, such as benefits administrator, associated with specified instances of Oracle Fusion Human Capital Management (HCM) data, such as one or more positions or all persons in a department.

Identity

A person representing a worker, supplier, or customer.

Interface table

A database table used for transferring data between applications or from an external application or data file.

Inventory organization

A logical or physical entity in the enterprise that is used to store definitions of items or store and transact items.

Inventory organization

An organization that tracks inventory transactions and balances, and can manufacture or distribute products.

Item cost profile

Cost accounting method assigned to an item. Items in a cost organization book can use different cost profiles. For example, you can use a cost profile to calculate average cost at the lot or serial level, and another cost profile to calculate average cost at the cost organization level.

Item master

A collection of data that describes items and their attributes recorded in a database file.

Item organization

Item definition where inventory balances are not stored and movement of inventory is not tracked in the applications. Item attributes that carry financial and accounting information are hidden.

Item subinventory

An association of an item with a subinventory that is created when you add an item to a subinventory.

Item validation organization

An inventory organization whose primary or secondary unit of measure is used as the costing unit of measure for the item in the cost organization to which that inventory organization belongs. The item master organization can also be designated as the item validation organization.
Items
Entries within the Product master database. For example, items for a manufacturing company can include nuts, bolts, and screws.

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.

job role
A role for a specific job consisting of duties, such as an accounts payable manager or application implementation consultant. A type of enterprise role.

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.

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.

key flexfield segment instance
A single occurrence of a key flexfield segment in a key flexfield structure instance.

key flexfield structure
The arrangement of segments in a key flexfield. In some cases, multiple structures can be defined for a single key flexfield.

key flexfield structure instance
A single occurrence of a key flexfield structure that shares the same order of segments as every other instance of the key flexfield structure, but uses different value sets to validate the segments.
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 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.

lookup code
A value available for lookup within a lookup type such as the code BLUE within the lookup type COLORS.

lookup type
A set of lookup codes to be used together as a list of values on a field in the user interface.

mainline
A branch of data that serves as a single source of truth.

manufacturing facilities
Employed in the making of goods for sale such as a factory or plant.

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.

OWLCS
Abbreviation for Oracle WebLogic Communication Services. Offers the TPCC service to Fusion applications and sets up the calls via SIP integration with the telephony network.

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.

pending worker
A person who will be hired or start a contingent worker placement and for whom you create a person record that is effective before the hire or start date.

perpetual average cost
The average cost of an item, derived by continually averaging its valuation after each incoming transaction. The following equation always holds for each item:

\[ \text{average cost of item} = \frac{\text{sum of debits and credits in inventory general ledger balance}}{\text{on-hand quantity}}. \]

person type
A subcategory of a system person type, which the enterprise can define. Person type is specified for a person at the employment-terms or assignment level.

personally identifiable information
Any piece of information that can potentially be used to uniquely identify, contact, or locate a single person. Within the context of an enterprise, some PII data can be considered public, such as a person's name and work phone number, while other PII data is confidential, such as national identifier or passport number.

PL/SQL
Abbreviation for procedural structured queried language.

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.
primary ledger
Main record-keeping ledger.

privilege
A grant or entitlement of access to functions and data. A privilege is a single, real world action on a single business object.

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.

profile option level
A level at which profile option values are defined. Site, product, and user are predefined levels.

profile option level hierarchy
The ordering of profile option levels. The order of the levels in the hierarchy determines which levels take precedence.

profile option value
The value portion of a profile option's name and value. A profile option may have multiple values set at different levels, such as site or user.

PSTN
Abbreviation for public switched telephone network which is the network of the world’s public circuit-switched telephone networks.

receipt layer
Unique identification of delivery or put away of an item into inventory.

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 group
A logical grouping of tables that correspond to logical entities such as payment terms defined across multiple tables or views. Grouping establishes common partitioning requirements across the entities causing them to share the same set assignments.

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.

RMA
Abbreviation for return material authorization.

role
Controls access to application functions and data.

role hierarchy
Structure of roles to reflect an organization’s lines of authority and responsibility. In a role hierarchy, a parent role inherits all the entitlement of one or more child roles.

role mapping
A relationship between one or more job roles, abstract roles, and data roles and one or more conditions. Depending on role-mapping options, the role can be provisioned to or by users with at least one assignment that matches the conditions in the role mapping.

role provisioning
The automatic or manual allocation of an abstract role, a job role, or a data role to a user.

sandbox
A runtime session that commits changes out of reach of mainline users.

security profile
A set of criteria that identifies one or more human capital management (HCM) objects of a single type for the purposes of securing access to those objects. Security profiles can be defined for persons, organizations, positions, countries, LDGs, document types, payrolls, payroll flows, and workforce business processes.

security reference implementation
Predefined function and data security in Oracle Fusion Applications, including role based access control, and policies that protect functions, data,
and segregation of duties. The reference implementation supports identity management, access provisioning, and security enforcement across the tools, data transformations, access methods, and the information life cycle of an enterprise.

**segregation of duties**

An internal control to prevent a single individual from performing two or more phases of a business transaction or operation that could result in fraud.

**service provider model**

A business unit that provides specific business functions for another business unit.

**set**

Reference data that is organized into groups appropriate to organizational entities, to enable reference data sharing.

**set enabled**

An entity, such as a lookup, customer, location, organization, or document attachment, that is allowed to participate in reference data sharing by drawing on the data of a reference data set.

**set-level definition**

Enables you to segment and share your reference data. Entities that are defined at the set level can be shared by all cost organizations belonging to that set. You can also use the **Common** set to share the same reference data across all cost organizations.

**SOA**

Abbreviation for service-oriented architecture.

**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.
SQL predicate
A type of condition using SQL to constrain the data secured by a data security policy.

storage facilities
Commercial building for storage of goods such as a warehouse.

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 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.

system person type
A fixed name that the application uses to identify a group of people.

territory
A legally distinct region that is used in the country field of an address.

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.

valuation structure

Defines inventory control attributes that are used to calculate the cost of an item. For example, the valuation structure of an item can be inventory organization and subinventory, or lot, or grade.

valuation unit

 Defines the set of values for the control attributes that are used to calculate the cost of an item. For example, valuation unit V1 is defined by cost organization A, item I1, and lot L1.

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 relationship

An association between a person and a legal employer, where the worker type determines whether the relationship is a nonworker, contingent worker, or employee work relationship.

workflow

An automated process in which tasks are passed from a user, a group of users, or the application to another for consideration or action. The tasks are routed in a logical sequence to achieve an end result.

XML filter

A type of condition using XML to constrain the data secured by a data security policy.