

# **Oracle® Fusion Applications Incentive Compensation Implementation Guide**

11g Release 6 (11.1.6)

**Part Number E20381-06**

September 2012

Oracle® Fusion Applications Incentive Compensation Implementation Guide

Part Number E20381-06

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

Authors: Lynn Raiser, Tina Brand, Angela Brown, Rakesh Chaudry, Essan Ni Jirman, Suzanne Kinkead, P. S. G. V. Sekhar, Kathryn Wohnoutka, Jacqui Wood

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

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

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

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

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

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

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

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

---

---

# Contents

## 1 Overview

Incentive Compensation Offering: Overview .....	1-1
Getting Started with an Implementation: Overview .....	1-2
Manage Application Implementation .....	1-3

## 2 Common Applications Configuration: Define Synchronization of User and Roles from LDAP

User and Role Synchronization: Explained .....	2-1
--	-----

## 3 Common Applications Configuration: Define Implementation Users

Initial Security Administration: Critical Choices .....	3-1
Initial Security Administration: Worked Example .....	3-3

## 4 Common Applications Configuration: Define Enterprise Structures for Incentive Compensation

Enterprise Structures: Overview .....	4-1
Enterprise Structures Business Process Model: Explained .....	4-3
Global Enterprise Configuration: Points to Consider .....	4-5
Modeling Your Enterprise Management Structure in Oracle Fusion: Example .....	4-5
Define Initial Configuration .....	4-8
Define Reference Data Sharing .....	4-35
Define Enterprise for Incentive Compensation .....	4-42
Define Legal Entities for Incentive Compensation .....	4-44
Define Chart of Accounts for Enterprise Structures for Incentive Compensation .....	4-49
Define Business Units for Incentive Compensation .....	4-64
Define Workforce Structures for Incentive Compensation .....	4-73

## 5 Common Applications Configuration: Define Persons for Incentive Compensation

Securing Identities and Users: Points To Consider .....	5-1
Role Provisioning and Deprovisioning: Explained .....	5-4
Role Mappings: Explained .....	5-6
Role Mappings: Examples .....	5-9

## 6 Common Applications Configuration: Define Trading Community Details for Incentive Compensation

Define Source Systems for Incentive Compensation .....	6-1
Define Geographies .....	6-2
Define Import for Incentive Compensation .....	6-16

## **7 Common Applications Configuration: Define Resources for Incentive Compensation**

Identifying Resources: Explained .....	7-1
Resource Directory: Explained .....	7-1
Resources: How They Work within a Team .....	7-2
Team Usage: Explained .....	7-2
Provisioning Security Roles: Explained .....	7-2
FAQs for Define Resources .....	7-2

## **8 Common Applications Configuration: Define Security for Incentive Compensation**

Security Tasks: Highlights .....	8-1
Defining Security After Enterprise Setup: Points to Consider .....	8-4
Security Tasks and Oracle Fusion Applications: How They Fit Together .....	8-8
Security Tasks: Overview .....	8-11
Define Data Security .....	8-15
Define Users .....	8-27

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

Segregation of Duties: Explained .....	9-1
Segregation of Duties in the Security Reference Implementation: Explained .....	9-2
Defining Segregation of Duties Policies: Points To Consider .....	9-4
Managing Segregation of Duties Risks and Violations: Critical Choices .....	9-6
Role Provisioning and Segregation of Duties: How They Work Together .....	9-8

## **10 Common Applications Configuration: Define Approval Management for Incentive Compensation**

Approval Management: Highlights .....	10-1
Approval Management Configuration Options for Oracle Fusion Incentive Compensation: Explained .....	10-2
Incentive Compensation Payment Approval Process: How It Is Configured .....	10-3
Dispute Management and Approval: How It Is Configured .....	10-6
Manage Task Configurations for Incentive Compensation .....	10-7

## **11 Common Applications Configuration: Define Help Configuration**

Define Help Configuration: Overview .....	11-1
Set Help Options .....	11-1
FAQs for Assign Help Text Administration Duty .....	11-4
Manage Help Security Groups .....	11-4

Help File Customization .....	11-5
Embedded Help Customization .....	11-16

## 12 Common Applications Configuration: Define Application Toolkit Configuration

Define Application Toolkit Configuration: Overview .....	12-1
FAQs for Map Reports to Work Areas .....	12-1
Set Watchlist Options .....	12-2

## 13 Common Applications Configuration: Maintain Common Reference Objects

Maintain Common Reference Objects: Overview .....	13-1
Define Application Taxonomy .....	13-1
Define Reference Data Sharing .....	13-4
Define Lookups .....	13-8
Manage Messages .....	13-15
Define ISO Reference Data .....	13-17
Manage Data Security Policies .....	13-20
Define Document Sequences .....	13-29
Define Trees .....	13-33
Define Profile Options .....	13-48
Define Flexfields .....	13-57
Define Attachments .....	13-100
Set Activity Stream Options .....	13-104
Manage Menu Customizations .....	13-104
Manage Applications Core Common Reference Objects: Manage Applications Core Messages .....	13-105
Manage Applications Core Common Reference Objects: Manage Applications Core Administrator Profile Values .....	13-106

## 14 Define Incentive Compensation Shared Configuration

Manage Profile Options and Flexfields .....	14-1
Incentive Compensation Flexfields: Explained .....	14-1
Manage Calendars .....	14-1
Manage Currencies and Currency Conversions .....	14-4

## 15 Define Incentive Compensation Custom Qualifiers and Lookups

Manage Incentive Compensation Custom Qualifiers and Lookups: Explained .....	15-1
--	------

## 16 Define Incentive Compensation Business Unit Configuration

Manage Intervals .....	16-1
Configure Tables and Columns .....	16-3
Manage Earning Types .....	16-18
Manage Open Period Process .....	16-19

Manage Participant Import .....	16-22
---------------------------------	-------

## **17 Define Transactional Business Intelligence Configuration**

Define Transactional Business Intelligence Configuration: Highlights .....	17-1
Access to Person Data .....	17-1

## **18 Define Extensions and Page Customizations**

Define Custom Enterprise Scheduler Jobs .....	18-1
Customization and Sandboxes .....	18-3
FAQs for Page Customizations .....	18-6

## **19 Importing and Exporting Setup Data**

Configuration Packages: Explained .....	19-1
Exporting and Importing Setup Data: Explained .....	19-1
Moving Common Reference Objects .....	19-2

---

---

# 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.

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

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

## Other Information Sources

### My Oracle Support

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

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

### Oracle Enterprise Repository for Oracle Fusion Applications

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



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

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

## Documentation Accessibility

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

## Comments and Suggestions

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



---

# Overview

## Incentive Compensation Offering: Overview

Using the Manage Incentive Compensation business process area, your enterprise can create complex calculations for nondiscretionary incentive plans such as sales commission plans or bonuses with formulaic calculations. It can also scale incentive plans for all organizations, measure performance to metrics, reward appropriate behavior, and analyze the impact of incentive programs on financial and sales goals.

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 you can select when you configure the offering, and lists of business objects and enterprise applications 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.

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

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

- Define Common Applications Configuration for Incentive Compensation
- Define Incentive Compensation
- Define Transactional Business Intelligence Configuration
- Define Extensions for Incentive Compensation

### Define Common Applications Configuration

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

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

## **Define Incentive Compensation Task List**

This task list contains tasks to set up incentive compensation shared and business unit objects, including managing calendars, currency conversions, parameters, intervals, and earning types as well as configuring tables and columns.

## **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 Incentive Compensation**

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**

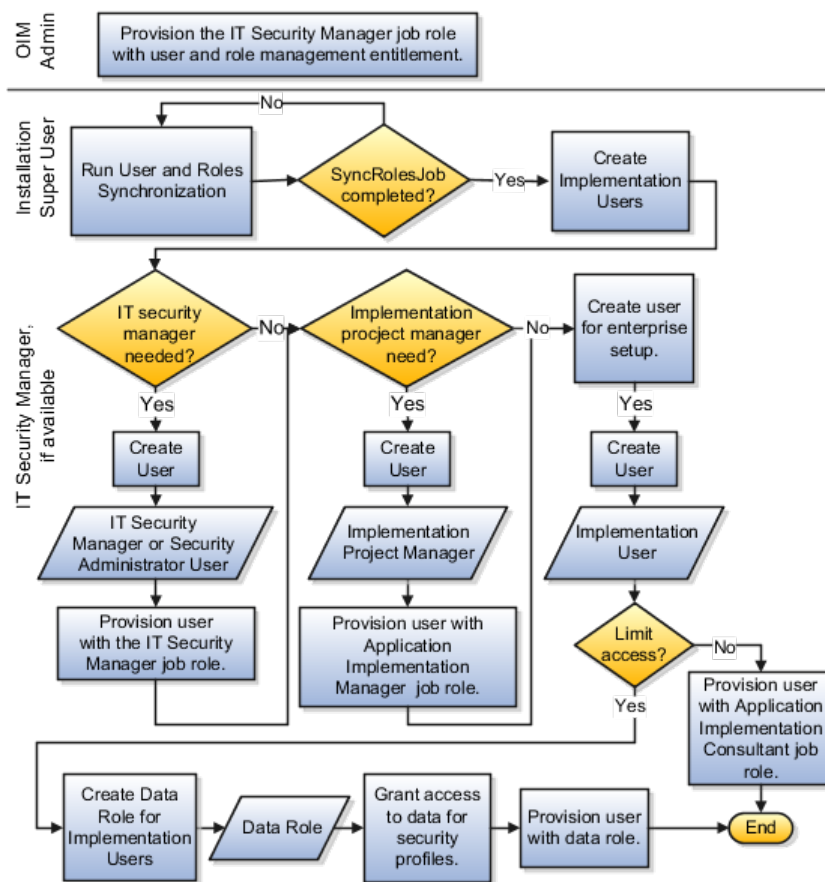
To start an Oracle Fusion Applications implementation, you must setup one or more initial users using the super user that was created during installation and provisioning of the Oracle Fusion Applications environment. Because Oracle Fusion Applications is secure as delivered, the process of enabling 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.
  - a. Synchronize users and roles in the Lightweight Directory Access Protocol (LDAP) store with HCM user management by using the Run User and Roles Synchronization Process task.
  - 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
- Workforce Deployment Implementation Guide
- Workforce Development Implementation Guide
- Incentive Compensation Implementation Guide
- Procurement Implementation Guide
- P6 EPPM Administrator's Guide for an Oracle Database
- P6 EPPM Administrator's Guide for Microsoft SQL Server Database

## Implementation Projects: Explained

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

### Maintaining Setup Data

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

### Exporting and Importing

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

### Selecting Offerings

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

## Offerings: Explained

Offerings are application solution sets representing one or more business processes and activities that you typically provision and implement as a unit.

They are, therefore, the primary drivers of functional setup of Oracle Fusion applications. Some of the examples of offerings are Financials, Procurement, Sales, Marketing, Order Orchestration, and Workforce Deployment. An offering may have one or more options or feature choices.

### **Implementation Task Lists**

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

### **Enabling Offerings**

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

### **Provisioning Offerings**

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

## **Options: Explained**

Each offering in general includes a set of standard functionality and a set of optional modules, which are called options. For example, in addition to standard Opportunity Management, the Sales offering includes optional functionality such as Sales Catalog, Sales Forecasting, Sales Prediction Engine, and Outlook Integration. These optional functions may not be relevant to all application implementations. Because these are subprocesses within an offering, you do not always implement options that are not core to the standard transactions of the offering.

## **Feature Choices: Explained**

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



You can configure feature choices in three different ways:

### **Yes or No**

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

### **Single Select**

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

### **Multi-Select**

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



---

## Common Applications Configuration: Define Synchronization of User and Roles from LDAP

### 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.

---

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

---

#### Tip

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.

---



---

## Common Applications Configuration: Define Implementation Users

### 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.

Decision	In this Example
How to sign in to Oracle Fusion Applications for the first time	Use the super user account that was created when installing and provisioning Oracle Fusion Applications (for example, FAADMIN).
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	Perform the Run User and Roles Synchronization Process task.
How to create a first implementation user	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.
How to establish security administration users	Define an IT security manager user provisioned with the IT Security Manager job role.
How to establish an implementation user with access to set up enterprise structures	Define an implementation user provisioned with the Application Implementation Consultant job role.

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:

Attribute	Value	Example
Last name	<any valid string>	Smith
Organization	Xellerate Users	N/A
User type	Non Worker	N/A
User login	<any valid string>	IT_SECURITY_MANAGER
Login password	<any valid string>	SeKur1TyPa\$\$w0Rd

---

### 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 too set up all enterprise structures.

---



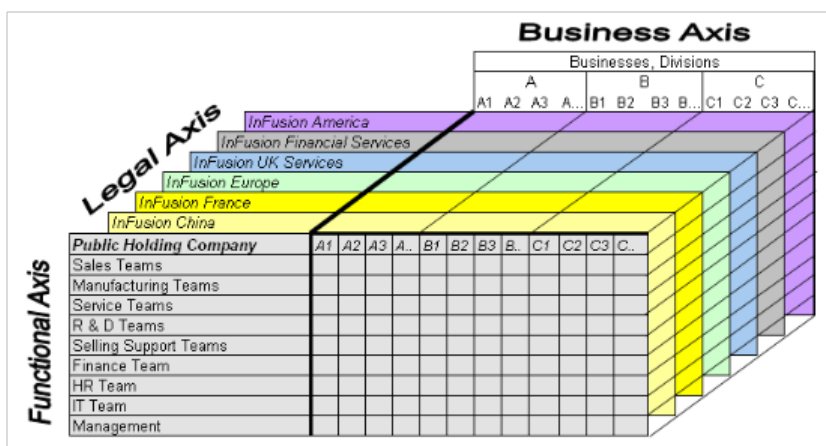
# Common Applications Configuration: Define Enterprise Structures for Incentive Compensation

## Enterprise Structures: Overview

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

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

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



## Legal Structure

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

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

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

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

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

## Management Structure

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

## Functional Structure

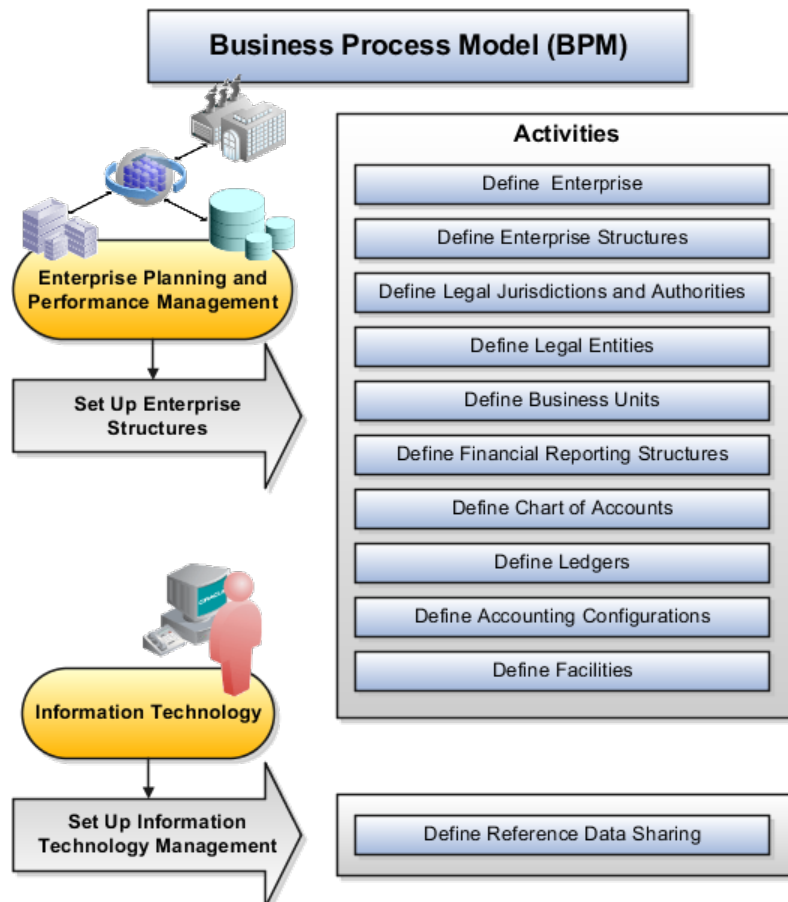
Straddling the legal and business organizations is a functional organization structured around people and their competencies. For example, sales,

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

## Enterprise Structures Business Process Model: Explained

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

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



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

---

#### Note

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



## Global Enterprise Configuration: Points to Consider

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

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

### Enterprise Configuration

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

### Business Unit Management

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

### Security Structure

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

### Compliance Requirements

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

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

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

### Scenario

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

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

### **InFusion Corporation**

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

### **Analysis**

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

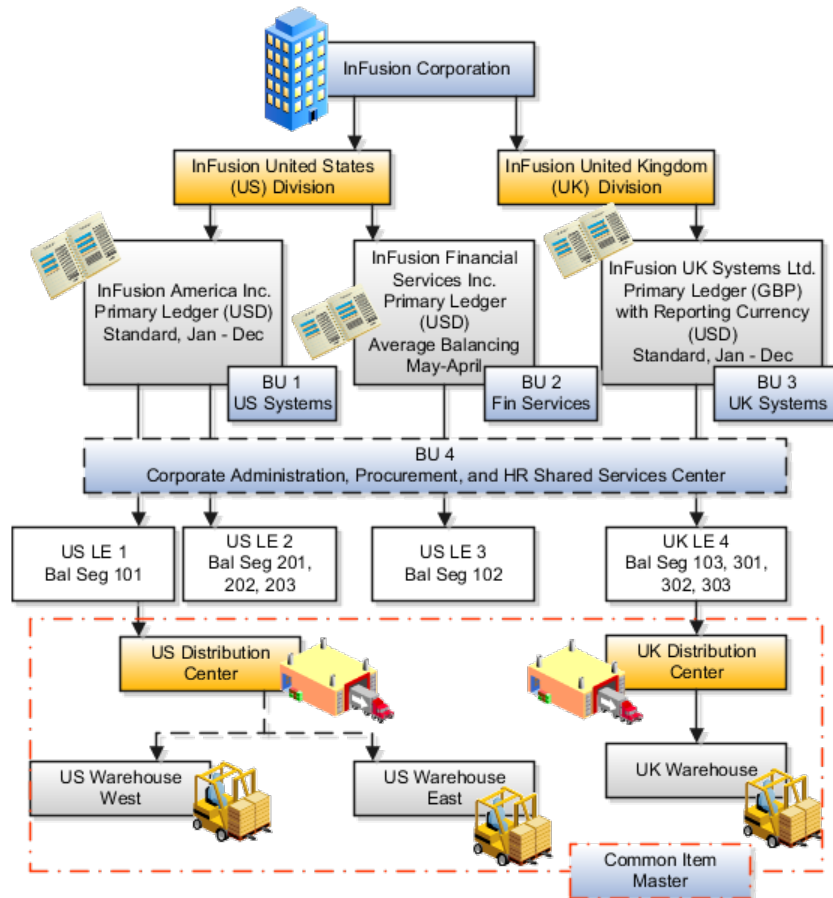
- Your company is required to report using US Generally Accepted Accounting Principles (GAAP) standards and UK Statements of Standard Accounting Practice and Financial Reporting Standards. How many ledgers do you need to achieve proper statutory reporting?
- Your managers need reports that show profit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

## **Global Enterprise Structure Model**

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

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

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



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

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

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

---

**Note**

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

---

---

**Note**

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

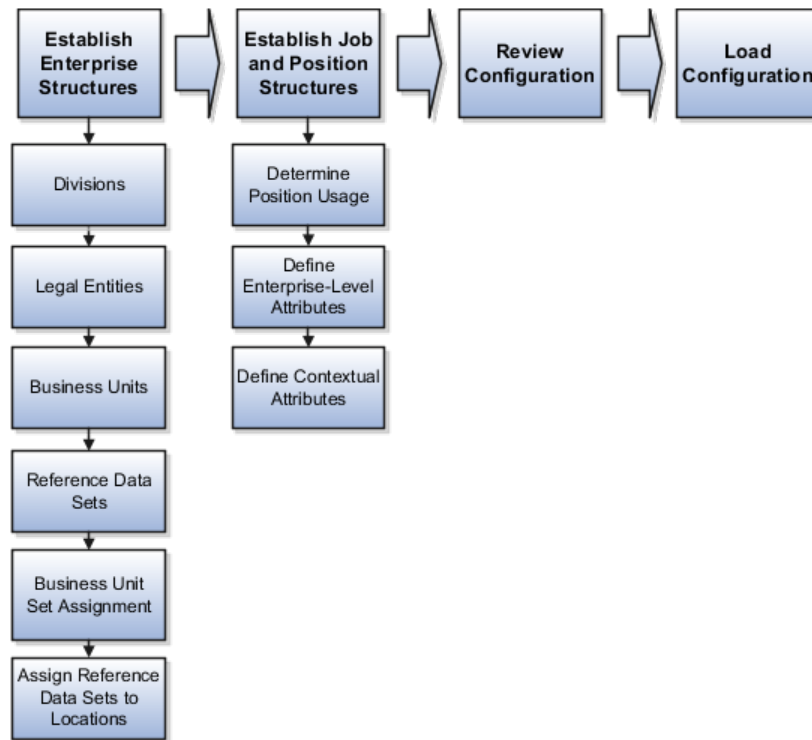
---

## Define Initial Configuration

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

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

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



To be able to use the Enterprise Structures Configurator, you must select the Enterprise Structures Guided Flow feature for your offerings on the Configure Offerings page in the Setup and Maintenance work area. If you do not select this feature, then you must set up your enterprise structure using individual tasks provided elsewhere in the offerings, and you cannot create multiple configurations to compare different scenarios.

### Establish Enterprise Structures

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

### Establish Job and Position Structures

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

## Review Configuration

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

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

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

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

---

### Note

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

---

## Load Configuration

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

# Designing an Enterprise Configuration: Example

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

## Scenario

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

## Enterprise Details

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

## Analysis

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

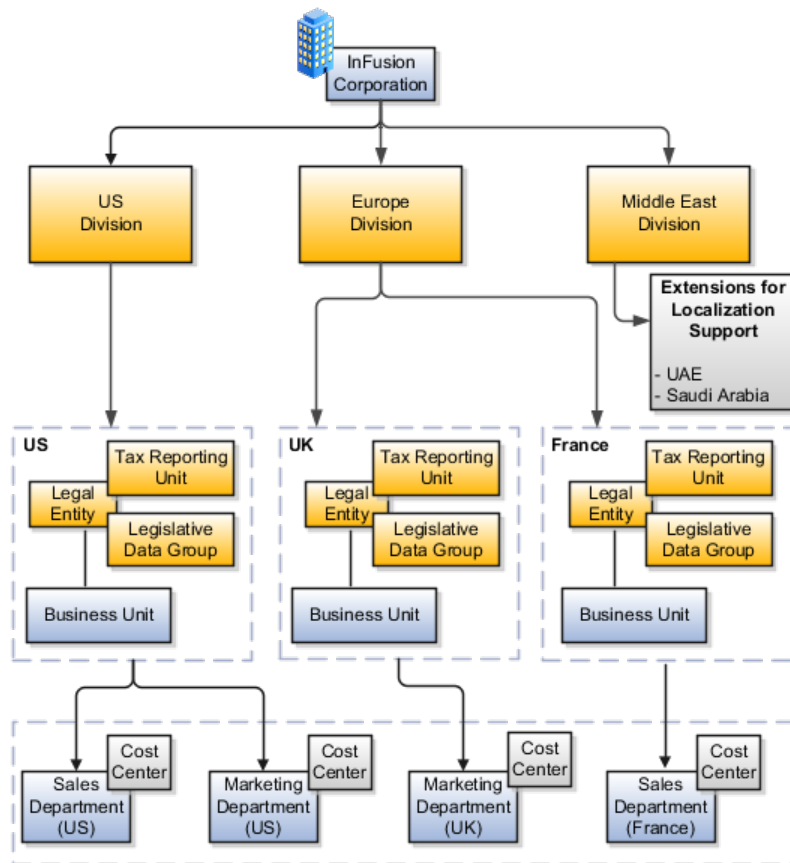
InFusion requires legal entities with legal employers, payroll statutory units, tax reporting units, and legislative data groups for the US, UK, France, Saudi Arabia, and UAE, in order to employ and pay its workers in those countries.

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

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

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

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



## Division: Explained

Managing multiple businesses requires that you segregate them by their strategic objectives and measure their results. Responsibility to reach objectives can be delegated along the management structure. Although related to your legal

structure, the business organizational hierarchies do not need to reflect directly the legal structure of the enterprise. The management entities and structure can include divisions and subdivisions, lines of business, and other strategic business units, and include their own revenue and cost centers. These organizations can be included in many alternative hierarchies and used for reporting, as long as they have representation in the chart of accounts.

## **Divisions**

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

A division is a profit center or grouping of profit and cost centers, where the division manager is responsible for attaining business goals including profit goals. A division can be responsible for a share of the company's existing product lines or for a separate business. Managers of divisions may also have return on investment goals requiring tracking of the assets and liabilities of the division. The division manager reports to a top corporate executive.

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

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

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

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

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

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

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

### **Scenario**

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



## Details of the Acquired Company

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

## Analysis

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

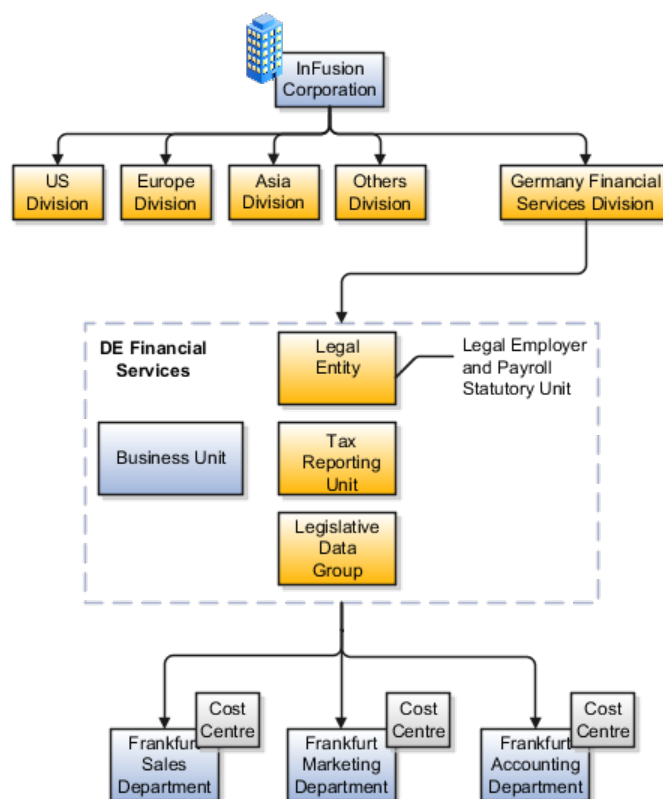
Decision to Consider	In This Example
Create location?	The financial services company is based in Frankfurt as are the departments, so you need to create only one location.
Create separate division?	Yes. Although the new division will exist within the current enterprise structure, you want to keep the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

Create legislative data group?	Yes. Because you currently do not employ or pay people in Germany, you must create one legislative data group to run payroll for the workers in Germany.
--------------------------------	--

Based on the analysis, you must create the following:

- One new division
- One new location
- Three new departments
- Three new cost centers
- One new legal entity
- One new legislative data group

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



## Legal Entities: Explained

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

Legal entities have the right to own property, the right to trade, the responsibility to repay debt, and the responsibility to account for themselves to regulators, taxation authorities, and owners according to rules specified in the relevant legislation. Their rights and responsibilities may be enforced through the

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

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

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

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

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

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

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

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

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

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

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

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

## Automatically Creating Legal Entities

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

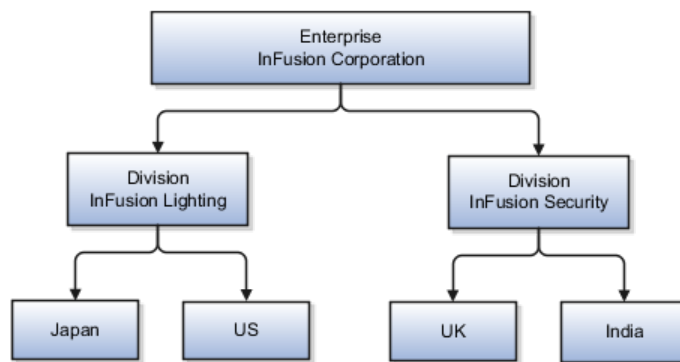
The application automatically creates the legal entities that you select, and identifies them as payroll statutory units and legal employers. For each country that you indicated that your enterprise operates in, and for each country that you created a location for, the application also automatically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

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

### Example: Creating Legal Entities Automatically

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

This figure illustrates InFusion Corporation's enterprise structure.



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

Country	Enterprise	InFusion Lighting	InFusion Security
Japan	No	Yes	No
US	No	Yes	No

UK	No	No	Yes
India	No	No	Yes

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

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

### Creating Legal Entities Using a Spreadsheet

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

## Legal Entity in Oracle Fusion: Points to Consider

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

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

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

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

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

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

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

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

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

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

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

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

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

One of your major responsibilities is to file financial statements for your legal entities. Map legal entities to specific ledgers using the Oracle Fusion General Ledger Accounting Configuration Manager. Within a ledger, you can optionally map a legal entity to one or more balancing segment values.

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

Oracle Fusion General Ledger supports up to three balancing segments. Best practices recommend that one of these segments represents your legal entity to

ease your requirement to account for your operations to regulatory agencies, tax authorities, and investors. Accounting for your operations means you must produce a balanced trial balance sheet by legal entity. If you account for many legal entities in a single ledger, you must:

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

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

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

---

**Note**

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

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

---

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

---

**Note**

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

you will not be able to distinguish them using the intercompany functionality or track their individual equity.

---

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

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

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

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

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

---

#### **Note**

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

---

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

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

### **Legal Entity and Payroll Reporting**

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

## **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

Management (HCM) security profiles to administer security for employees in business units.

## 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.

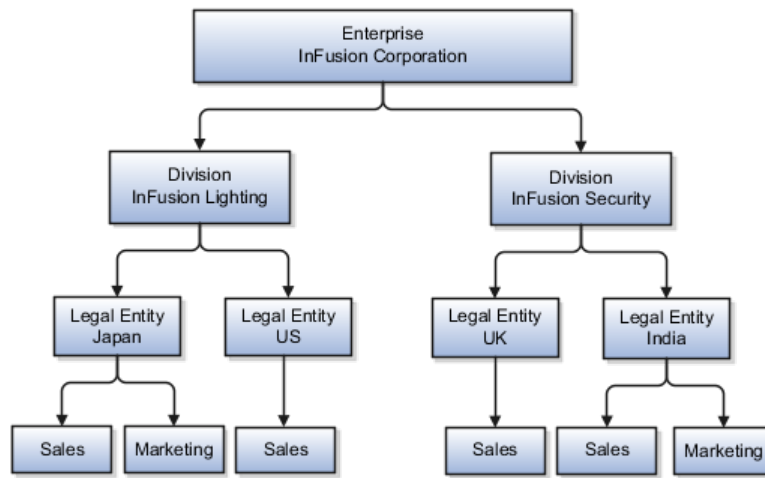
The business units listed by the application are suggestions only, and are meant to simplify the process to create business units. You are not required to select all of the business units suggested. When you navigate to the next page in the ESC guided flow, which is the Manage Business Units page, you 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.

Business Unit Level	Suggested Business Units
Country	<ul style="list-style-type: none"> <li>• US</li> <li>• UK</li> <li>• Japan</li> <li>• India</li> </ul>
Country and Division	<ul style="list-style-type: none"> <li>• InFusion Lighting: Japan</li> <li>• InFusion Lighting: US</li> <li>• InFusion Security: UK</li> <li>• InFusion Security: India</li> </ul>
Country and business function	<ul style="list-style-type: none"> <li>• Sales: Japan</li> <li>• Marketing: Japan</li> <li>• Sales: US</li> <li>• Sales: UK</li> <li>• Marketing: India</li> <li>• Sales: India</li> </ul>

Division	<ul style="list-style-type: none"> <li>• InFusion Lighting</li> <li>• InFusion Security</li> </ul>
Division and Legal Entity	<ul style="list-style-type: none"> <li>• InFusion Lighting: Japan</li> <li>• InFusion Lighting: US</li> <li>• InFusion Security: UK</li> <li>• InFusion Security: India</li> </ul>
Division and Business Function	<ul style="list-style-type: none"> <li>• InFusion Lighting, Sales</li> <li>• InFusion Lighting, Marketing</li> <li>• InFusion Security, Sales</li> <li>• InFusion Security, Marketing</li> </ul>
Business Function	<ul style="list-style-type: none"> <li>• Sales</li> <li>• Marketing</li> </ul>
Legal Entity	<ul style="list-style-type: none"> <li>• Legal Entity: Japan</li> <li>• Legal Entity: US</li> <li>• Legal Entity: UK</li> <li>• Legal Entity India</li> </ul>
Legal Entity and Business Function	<ul style="list-style-type: none"> <li>• Legal Entity: Japan, Sales</li> <li>• Legal Entity: Japan, Marketing</li> <li>• Legal Entity: US, Sales</li> <li>• Legal Entity: UK, Sales</li> <li>• Legal Entity India, Marketing</li> <li>• Legal Entity India, Sales</li> </ul>

### Manually Creating Business Units

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.

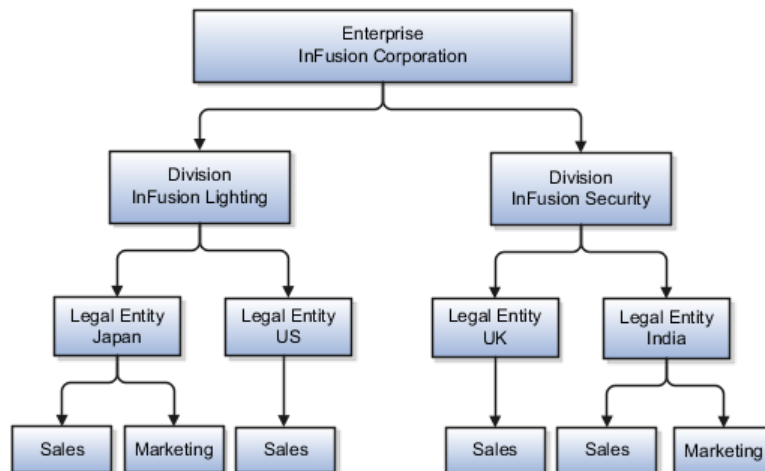
### Business Unit Set Assignment

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:

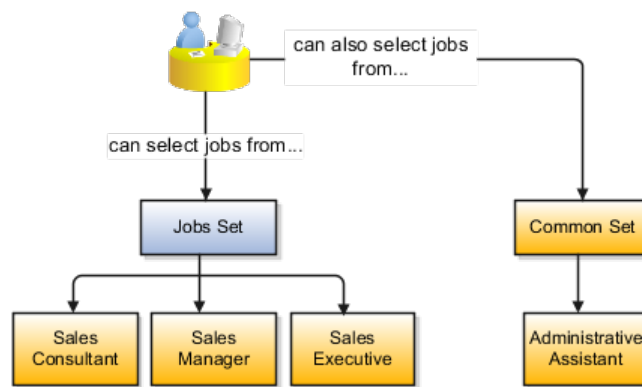
Business Unit	Default Set Assignment	Set Assignment Overrides
Sales_Japan	Sales_Japan_Set for grades, departments, and locations	Jobs set for jobs
Marketing_Japan	Mktg_Japan_Set for grades, departments, and locations	None
Sales_US	Sales_US_Set for grades, departments, and locations	Jobs set for jobs
Sales_UK	Sales_UK_Set for grades, departments, and locations	Jobs set for jobs
Marketing_India	Mktg_India_Set for grades, departments, and locations	None
Sales_India	Sales_India_Set for grades, departments, and locations	Jobs set for jobs

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.

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

### 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.

Industry	We always replace employees by rehiring to same role	We replace the head count, but the manager can use the head count in a different job	We rehire to the same position, but the manager can request a reallocation of budget to a different post
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.)	Positions	Jobs	Jobs
Controlled (An industry that is highly structured in which all aspects of work and remuneration are well organized and regulated.)	Positions	Positions	Positions
Manufacturing	Positions	Jobs	Positions
Retail	Positions	Jobs	Positions
Education	Positions	Jobs	Positions
Other	Positions	Jobs	Jobs

## 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.

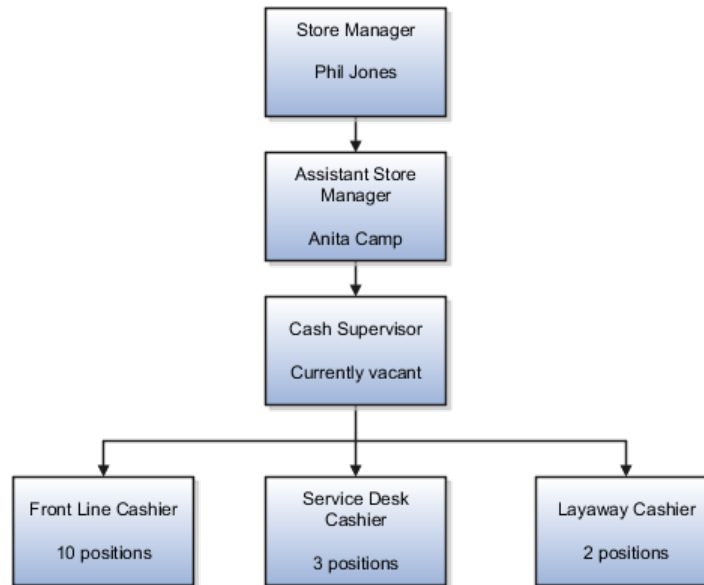
### Retail Industry

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.

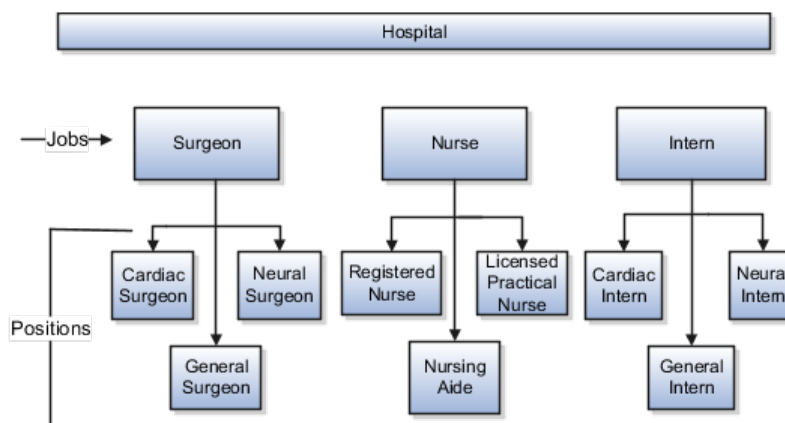


## Health Care Industry

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

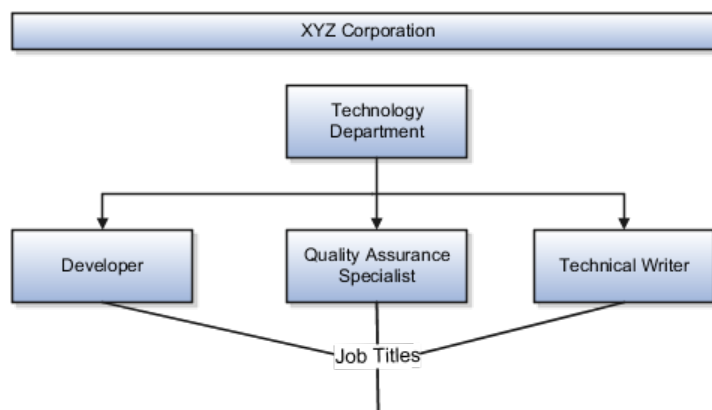
Jobs are typically used without positions by service industries where flexibility and organizational change are key features.

## 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.



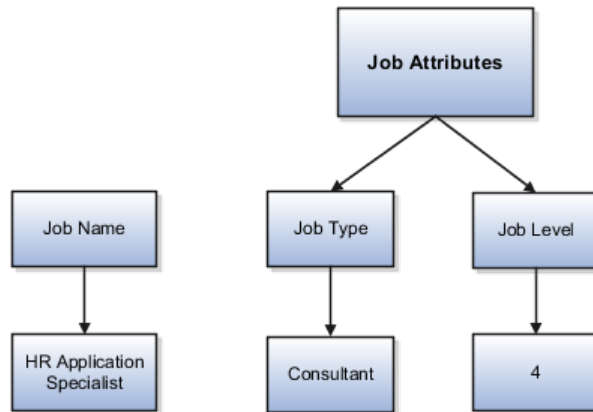
## Job and Position Structures: Explained

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.

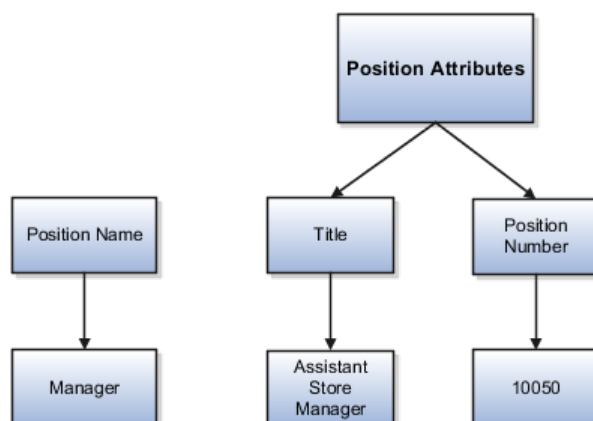
This figure illustrates how job type and job level provide further details for the HR Application Specialist job.



### 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.

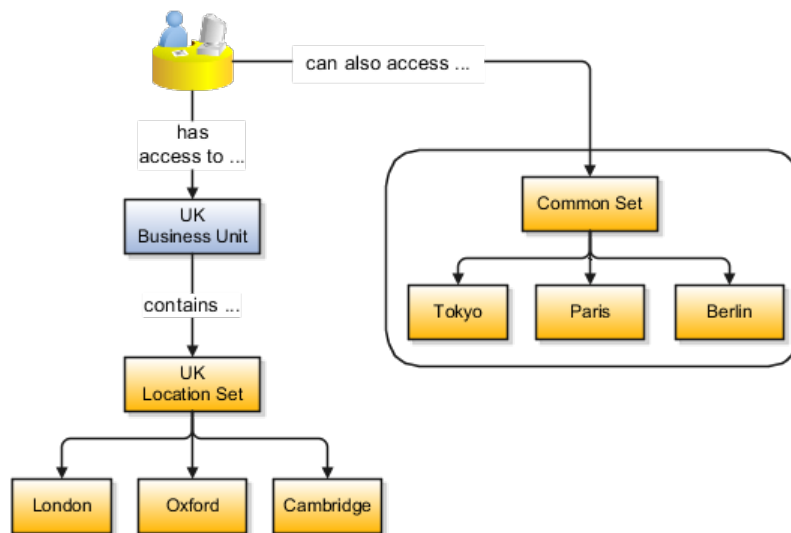
Your enterprise can decide that some aspects of corporate policy should affect all business units and leave other aspects to the discretion of the business unit manager. This allows your enterprise to balance autonomy and control for each business unit. For example, if your enterprise holds business 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 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.

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

### 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 and supplier sites are two such product specific reference data objects that use product specific mechanisms to share 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

- Procurement configuration data

## 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.

Application Name	Reference Data Object	Method of Sharing
Trading Community Model	Customer Account Relationship	Assignment to one set only, with common values
Trading Community Model	Customer Account Site	Assignment to one set only, with common values
Trading Community Model	Sales Person	Assignment to one set only, with common values
Opportunity Management	Sales Method Group	Assignment to one set only, with common values
Work Management	Assessment Templates	Assignment to one set only, with common values
Enterprise Contracts	Contract Types	Assignment to one set only, with common values
Sales	Sales Method	Assignment to one set only, with common values
Common Components	Activity Templates	Assignment to one set only, with common values
Payables	Payment Terms	Assignment to multiple sets, no common values allowed
Receivables	Accounting Rules	Assignment to one set only, with common values
Receivables	Aging Buckets	Assignment to one set only, with common values
Receivables	Auto Cash Rules	Assignment to one set only, with common values
Receivables	Collectors	Assignment to one set only, with common values
Receivables	Lockbox	Assignment to one set only, with common values
Receivables	Memo Lines	Assignment to one set only, with common values
Receivables	Payment Terms	Assignment to one set only, with common values
Receivables	Remit To Address	Assignment to one set only, with common values
Receivables	Revenue Contingencies	Assignment to one set only, with common values
Receivables	Transaction Source	Assignment to one set only, with common values

Receivables	Transaction Type	Assignment to one set only, with common values
Advanced Collections	Collections Setups	Assignment to one set only, with common values
Advanced Collections	Dunning Plans	Assignment to one set only, with common values
Tax	Tax Classification Codes	Assignment to one set only, with common values
Performance Management	Performance Templates	Assignment to one set only, with common values
Human Resources	Departments	Assignment to one set only, with common values
Human Resources	Jobs	Assignment to one set only, with common values
Human Resources	Locations	Assignment to one set only, with common values
Human Resources	Grades	Assignment to one set only, with common values
Project Billing	Project and Contract Billing	Assignment to multiple sets, common values not allowed
Project Foundation	Project Accounting Definition	Assignment to one set only, no common values allowed
Project Foundation	Project Rates	Assignment to one set only, with common values
Distributed Order Orchestration	Hold Codes	Assignment to one set only, with common values
Distributed Order Orchestration	Orchestration Process	Assignment to one set only, with common values

### 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.

Application Name	Reference Data Object	Method of Sharing
Assets	Bonus Rules	Assignment to one set only, no common values allowed
Assets	Depreciation Ceilings	Assignment to one set only, no common values allowed
Assets	Depreciation Methods	Assignment to one set only, with common values
Assets	Asset Descriptions	Assignment to one set only, no common values allowed
Assets	Property Types	Assignment to one set only, with common values
Assets	Prorate Conventions	Assignment to one set only, no common values allowed

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.

Application Name	Reference Data Object	Method of Sharing
Cost Management	Cost Structure	Assignment to one set only, no common values allowed

### 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.

Application Name	Reference Data Object	Method of Sharing
Project Foundation	Project Definition	Assignment to multiple sets, no common values allowed
Project Foundation	Project Transaction Types	Assignment to multiple sets, no common values allowed

## Define Enterprise for Incentive Compensation

### 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.

## **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.

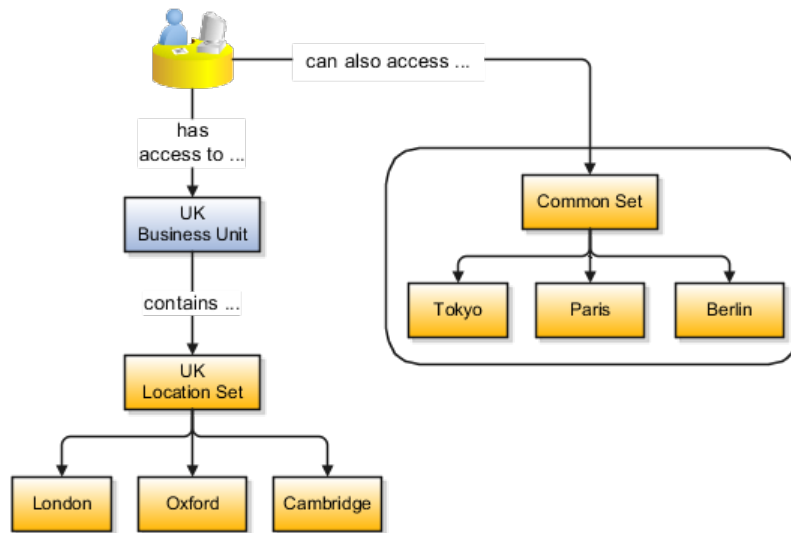
The locations that you create exist as separate structures that you can use for reporting purposes, and also in rules that determine employee eligibility for various types of compensation and benefits. You enter information about a location only once. Subsequently, when you set up other workforce structures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

## Define Legal Entities for Incentive Compensation

### 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.

---

**Note**

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

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

---

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

---

**Note**

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

---

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

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

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

Use Oracle Fusion Intercompany functionality for automatic creation of intercompany entries across your balancing segments. Intercompany processing

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

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

---

**Note**

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

---

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

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

**Legal Entity and Payroll Reporting**

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

## **Define Chart of Accounts for Enterprise Structures for Incentive Compensation**

### **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.

In some applications, different users need to see different segment structures for the same flexfield. A key flexfield can have multiple structures if registered to support more than one structure.

The flexfield can display different fields for different end users based on a data condition in your application data, such as the value of another field entered by the end user or the user's role. For example, the correctly formatted local postal address for customer service inquiries differs based on locale. A postal address key flexfield could display different segments and prompts for different end users based on a location condition in your application data, such as the user's role or a value entered by the user.

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.

Each structure may have one or more structure instances. 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.

## **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.

Level Of Dynamic Combination Creation	Controlled By:
Flexfield	Application development
Each usage or reference to the key flexfield	Application development
Structure instance	Administrators and implementation consultants
Other	Administrators and implementation consultants

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.

---

### Structure Delimiters

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.

### 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.

---

Enable segments to indicate that they are in use. A flexfield does not display disabled segments in runtime.

---

#### **Tip**

To protect the integrity of your data, disable a segment if you have already used it to enter data.

---

## **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.

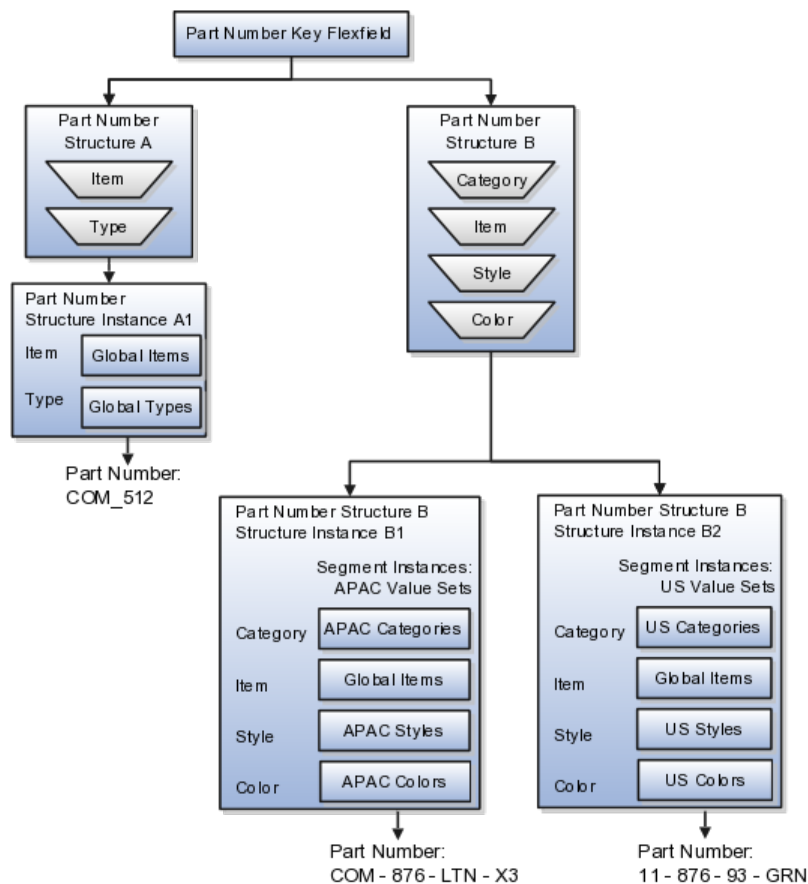
Differences among segment instances at the structure instance level include the following.

- 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

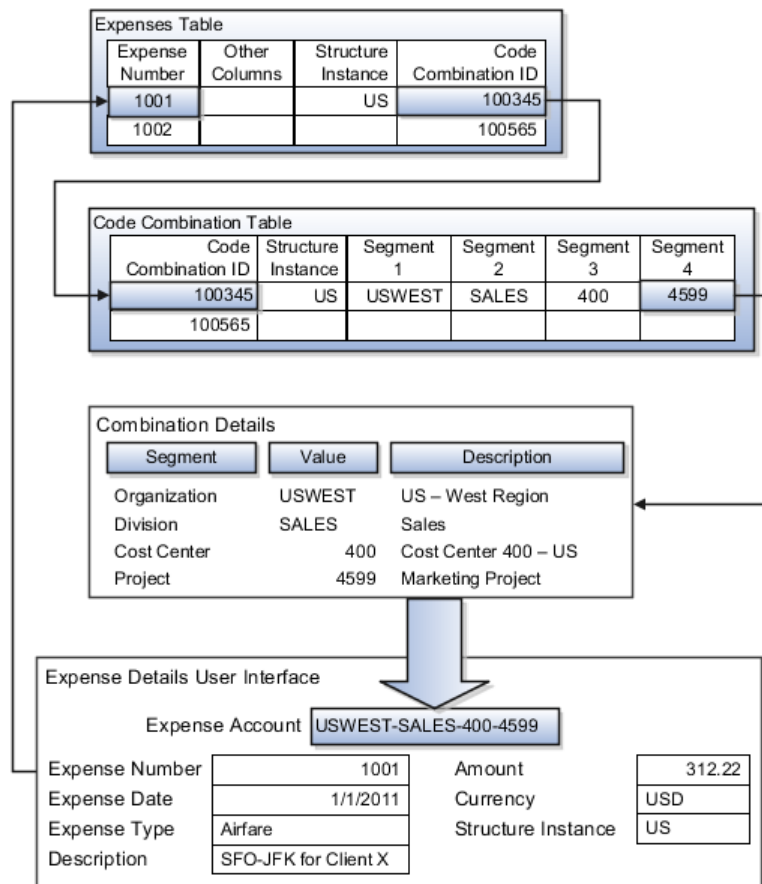
A user interface for entering expenses gives the user the option of selecting an expense account that identifies the cost center and other details needed for processing the expense.

## Analysis

The expense account field is a foreign key reference to a code combination (EXPENSE\_LINES.EXPENSE\_ACCOUNT = ACCOUNTS.CCID).

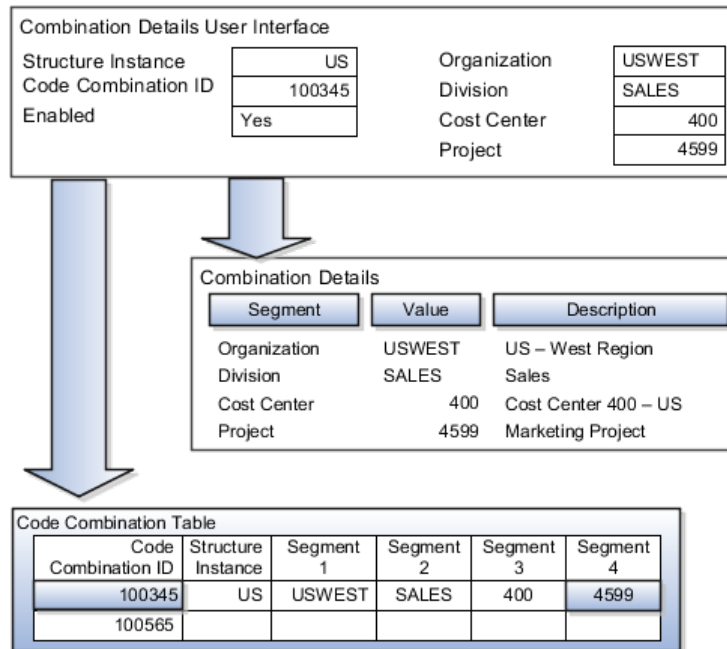
The code combination table supports entering account information, such as for expense accounts.

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.

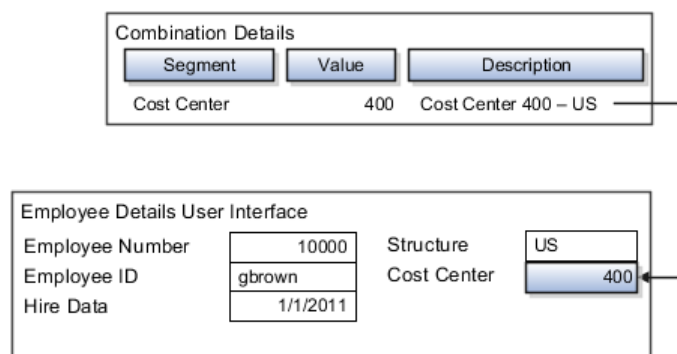


If dynamic combination creation is not enabled, then when entering an expense line, the user can only select an account that already exists in the ACCOUNTS (combinations) table. If they require an account that does not exist, they must consult with the appropriate application administrator who can add 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.



## 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.

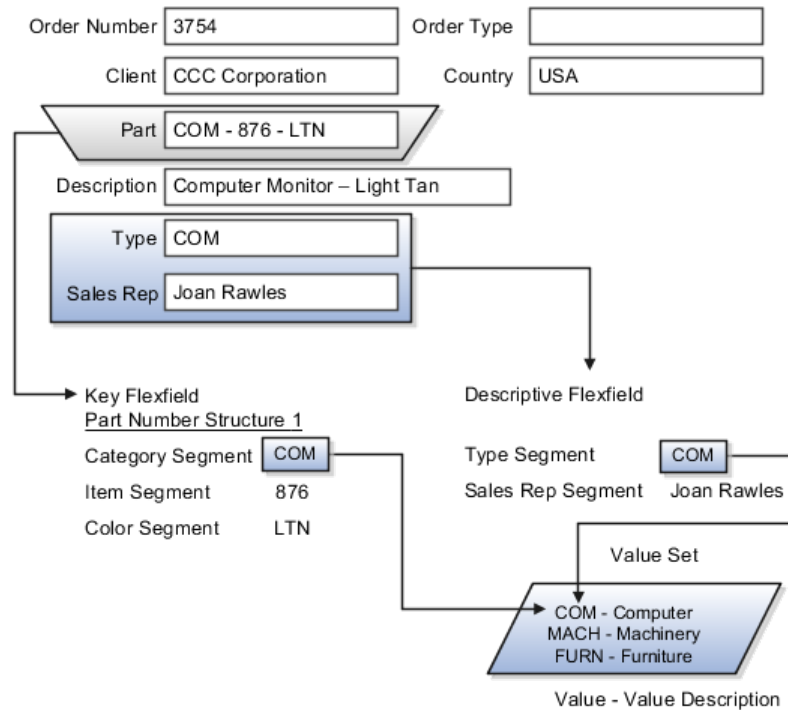
**Precision and Scale**

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.

# **Define Business Units for Incentive Compensation**

## **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

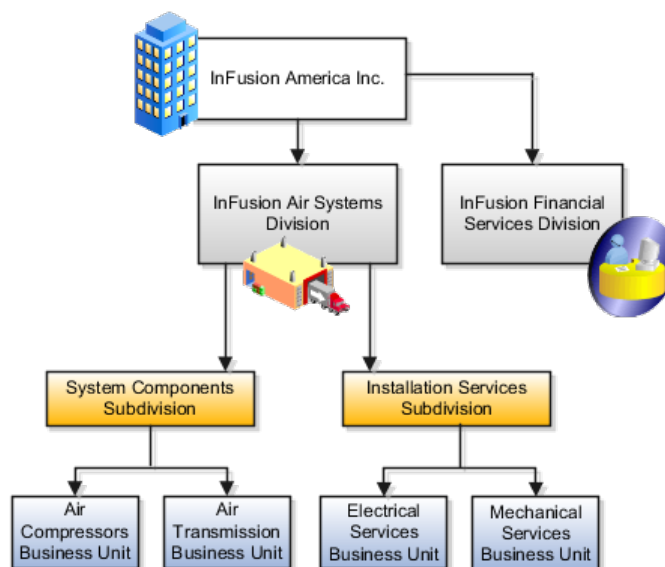
### Business Unit Hierarchy: Example

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.

## 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 Management (HCM) security profiles to administer security for employees in business units.

## **Setting Up Incentive Compensation Business Units: Points to Consider**

There are many factors that impact how you set up your incentive compensation business units for your global enterprise structure, including those dealing with incentive compensation plans as well as data security, processing, and reporting.

### **Incentive Compensation Plans**

Incentive compensation plan factors that impact how to best configure your incentive compensation business units include commonality across the organizational hierarchy as well as quantity and complexity of plans, for example:

- At which level do you use common incentive compensation plans: country, division, region, or global?
- Do your compensation plans use common components, expressions, and performance measures at the country, division, region or global level or is each plan independent?
- How many different compensation plans do you use per business unit?
- How complex are your compensation plans?
- Are your plans similar enough that you could use personalization of incentive plan data to handle the minor variations?

Oracle Fusion Incentive Compensation enables you to individualize many compensation plan values for participants, which can reduce the number of plans that you actually have to create and manage. Level of complexity also affects the quantity of plans you create, and in which business units it would be most efficient to create and maintain them.



## Data Security, Processing, and Reporting

Be sure to consider how you want to constrain data access and visibility as well as incentive processing, for example:

- At what level do you want to secure data: by line of business, division, country, or globally?
- Is your processing centralized, or do individual business units or regional centers perform the analyst function? Do they only work on their participants or is work pooled?
- How do you want to report on your business units and divisions?

Oracle Fusion Incentive Compensation supports processing across business units and teams. It also enables you to set a global operating currency and process incentive compensation in local currencies or using a global currency. You can introduce global sales teams and structures at any time, without changing your enterprise structure model.

## Incentive Compensation Business Units by Division and Zones: Example

This example uses a fictitious global company to demonstrate business unit analysis as part of enterprise structure planning. In this scenario, you are chairing a committee to create a model for your global enterprise structure.

### Enterprise Structure

You work for a multinational conglomerate that operates in 15 countries worldwide. You bought Oracle Fusion Incentive Compensation to implement as a standalone application.

You have three processing centers (Asia-Pacific, Americas, and Europe-Middle East) and two divisions:

- High-tech Products has 10 very complex compensation plans that are used globally. It creates all plans and administers quota worldwide in USD.
- Consumer Services has over 100 similar simple, but slightly variant locally-used compensation plans. It creates compensation plans with rate tiers and quotas in USD or EUR and pays all participants in their local currency.

There is no overlap of plans between the divisions. All employees report to in-region managers, there are no cross-region teams.

You want to segregate data for security purposes by division. Compensation analysts in different regional centers work on a 24 x 7 basis on paysheets for any participant worldwide in their assigned division. Both divisions and all business

units use USD as the operating currency so that executives can easily review all performance and expenses.

## Analysis

The following are elements to consider when creating the business units for your global enterprise structure.

- At which level do you use common incentive compensation plans: country, division, region, or global?
- Do your compensation plans use common components, expressions, and performance measures at the country, division, region or global level or is each plan independent?
- How many different compensation plans do you use per business unit? How complex are they?
- Are your plans similar enough that you could use personalization of incentive plan data to handle the minor variations?
- At what level do you want to secure data: by line of business, division, country, or globally?
- Is your processing centralized, or do individual business units or regional centers perform the analyst function? Do they only work on their participants or is work pooled?
- How do you want to report on your business units and divisions?

Because Oracle Fusion Incentive Compensation accommodates multicurrency processing and reporting, your committee recommends creating three separate business units.

- Set up one business unit to handle the 10 complex global compensation plans for the High-Tech Products division in one place.
- Set up the other two business units to handle the two zones (USD and EUR) for the Consumer Services division, with operating currency set to USD and EUR, respectively. This enables you to reduce the 100 differing plans to 5 global plans with personalized weights, quotas, and rates.

The implementation of three business units:

- Meets the currency processing requirements
- Provides consistent enforcement of company policies
- Improves efficiency across the organization

## Incentive Compensation Business Units by Region: Example

This example uses a fictitious global company to demonstrate business unit analysis as part of enterprise structure planning. In this scenario, you are chairing a committee to create a model for your global enterprise structure.

## Enterprise Structure

You work for a multinational conglomerate that operates in 15 countries worldwide. You bought Oracle Fusion Incentive Compensation to implement as a standalone application.

You have three processing centers (Asia-Pacific, Americas, and Europe-Middle East) and two divisions:

- Agricultural Products has three medium-complex compensation plans that are used globally.
- Consumer Products has over 50 simple plans, half of which are used globally and half regionally.

There is no overlap of plans between the divisions. You use local currency to create incentive compensation plan rate tiers and quotas, as well as to pay participants.

Employees may report to managers in different regions, who receive rollup credit. Also, employees in the Agricultural Products division may belong to sales teams with members from other regions.

You want to segregate data for security purposes by region; local analysts can work on compensation for either division, but only for participants in one region. Both divisions and all business units use local currency as the operating currency, so that executives can easily review all performance and expenses at the national level, across divisions and participants.

## Analysis

The following are elements to consider when creating the business units for your global enterprise structure.

- At which level do you use common incentive compensation plans: country, division, region, or global?
- Do your compensation plans use common components, expressions, and performance measures at the country, division, region or global level or is each plan independent?
- How many different compensation plans do you use per business unit? How complex are they?
- Are your plans similar enough that you could use personalization of incentive plan data to handle the minor variations?
- At what level do you want to secure data: by line of business, division, country, or globally?
- Is your processing centralized, or do individual business units or regional centers perform the analyst function? Do they only work on their participants or is work pooled?
- How do you want to report on your business units and divisions?

Because Oracle Fusion Incentive Compensation accommodates cross-region rollups and teams, your committee recommends creating three separate business units.

- Set up the three business units to correspond to the three processing center regions, combining operations for the Agricultural Products and Consumer Products divisions.
- For all business units, set the transaction currency to Participant home currency.
- Create and manage the three global plans for the Agricultural Products division and 25 global plans for the Consumer Products division in each of the three regional business units, as they are not too complex.

The implementation of three business units:

- Meets the currency processing requirements
- Provides consistent enforcement of company policies
- Improves efficiency across the organization

## **FAQs for Manage Business Units**

### **What's the difference between business units in Oracle E-Business Suite and Oracle Fusion Applications?**

In Oracle E-Business Suite, operating units are used to determine in which ledger a given subledger transaction is accounted and to partition setup reference data, processing and security.

In Oracle Fusion Applications, enable business units with all their business functions to replace your operating units in the Oracle E-Business Suite. Oracle Fusion Applications provide the additional functionality of assigning a manager to the business unit.

### **What's the difference between business units in PeopleSoft and Oracle Fusion Applications?**

PeopleSoft business units and Oracle E-Business Suite operating units have been combined to create the new Oracle Fusion Applications business unit functionality.

In PeopleSoft Enterprise, a business unit housed the configuration of only one business function.

A business unit can be configured for multiple business functions in Oracle Fusion Applications. The advantage is you no longer have to name multiple business units with the same name as you did in PeopleSoft Enterprise.

In PeopleSoft Enterprise, business units can be consolidated in a hierarchy. You can see the results of a single business unit or a set of business units. PeopleSoft Enterprise also allows you to produce financial statements for a business unit.

In Oracle Fusion Applications, this is accomplished by creating a business unit representation in a chart of accounts and building appropriate hierarchies.

## Define Workforce Structures for Incentive Compensation

### Cost Centers and Departments: Explained

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

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

- Cost centers
- Departments

#### Cost Centers

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

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

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

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

a balancing segment value in combination with the cost center values to report on a single legal entity.

## Departments

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

A manager of a department is typically responsible for:

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

---

### Note

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

---

The financial performance of departments is generally tracked through one or more cost centers. In Oracle Fusion Applications, departments are defined and classified as Department organizations. Oracle Fusion Human Capital Management (HCM) assigns workers to departments, and tracks the headcount at the departmental level.

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

## Department Classifications: Points to Consider

A department can be classified as a project organization, sales and marketing organization, or cost organization.

Oracle Fusion Human Capital Management (HCM) uses trees to model organization hierarchies. It provides seeded tree structures for department and other organizational hierarchies that can include organizations with any classification.

### Project Organization

Classify departments as a project owning organization to enable associating them with projects or tasks. The project association is one of the key drivers for project access security.

In addition, you must classify departments as project expenditure organizations to enable associating them to project expenditure items. Both project owning organizations and project expenditure organizations can be used by Oracle

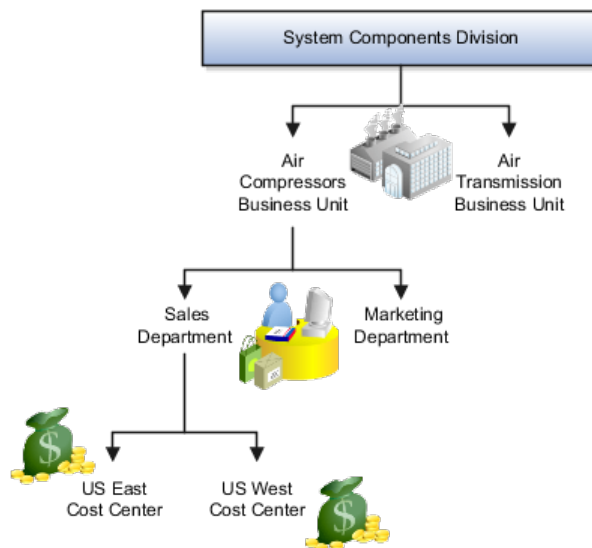
Fusion Subledger Accounting to derive accounts for posting Oracle Fusion Projects accounting entries to Oracle Fusion General Ledger.

## Sales and Marketing Organization

In Oracle Fusion Customer Relationship Management (CRM), you can define sales and marketing organizations. Sales organization hierarchies are used to report and forecast sales results. Sales people are defined as resources assigned to these organizations.

In some enterprises, the HCM departments and hierarchies correspond to sales organizations and hierarchies. It is important to examine the decision on how to model sales hierarchies in relationship to department hierarchies when implementing customer relationship management to eliminate any possible redundancy in the definition of the organizations.

The following figure illustrates a management hierarchy, in which the System Components Division tracks its expenses in two cost centers, Air Compressors and Air Transmission. At the department level, two organizations with a classifications of Department are defined, the Marketing Department and Sales Department. These two departments can be also identified as a Resource Organizations, which will allow assigning resources, such as sales people, and other CRM specific information to them. Each department is represented in the chart of accounts by more than one cost center, allowing for granular as well as hierarchical reporting.



## Cost Organization

Oracle Fusion Costing uses a cost organization to represent a single physical inventory facility or group of inventory storage centers, for example, inventory organizations. This cost organization can roll up to a manager with responsibility for the cost center in the financial reports.

A cost organization can represent a costing department. Consider this relationship when determining the setup of departments in HCM. There are no

system dependencies requiring these two entities, cost organization and costing department, be set up in the same way.

## Jobs: Explained

As part of your initial implementation, you specify whether to use jobs and positions, or only jobs. Jobs are typically used without positions by service industries where flexibility and organizational change are key features.

### Basic Details

Basic details for a job include an effective start date, a job set, a name, and a code.

A job code must be unique within a set. Therefore, you can create a job with the code DEV01 in the US set and another job with the same code in the UK set. However, if you create a job with the code DEV01 in the Common set, then you cannot create a job with the same code in any other set.

### Benchmark Information

You can identify a job as being a benchmark job. A benchmark job represents other jobs in reports and salary surveys. You can also select the benchmark for jobs. Benchmark details are for informational purposes only. A progression job is the next job in a career ladder.

### Progression Information

Progression jobs enable you to create a hierarchy of jobs and are used to provide the list of values for the Job field in the Promote Worker and Transfer Worker tasks. The list of values includes the next three jobs in the progression job hierarchy. For example, assume that you create a job called Junior Developer and select Developer as the progression job. In the Developer job, you select Senior Developer as the progression job. When you promote a junior developer, the list of values for the new job will include Developer and Senior Developer. You can select one of these values, or select another one.

### Jobs and Grades

You can assign grades that are valid for each job. If you are using positions, then the grades that you specify for the job become the default grades for the position.

### Evaluation Criteria

You can define evaluation criteria for a job, including the evaluation system, a date, and the unit of measure for the system. One predefined evaluation system is available, and that is the Hay system. An additional value of Custom is included in the list of values for the Evaluation System field, but you must add your own criteria and values for this system.

### Uploading Jobs Using a Spreadsheet

If you have a list of jobs 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 job information to the spreadsheet, and then upload



directly to your enterprise configuration. You can upload the spreadsheet multiple times to accommodate revisions.

## Jobs: Example

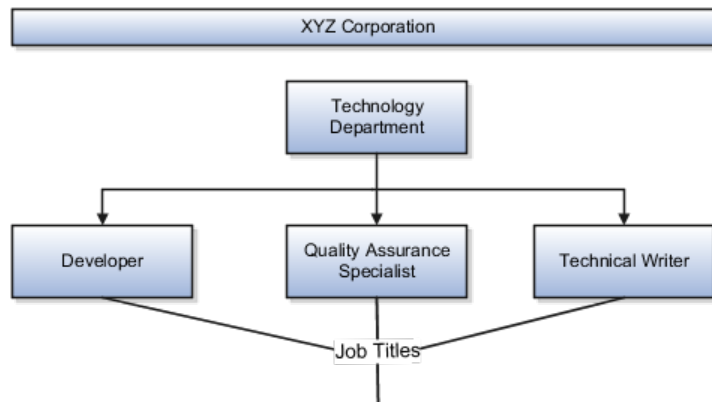
Jobs are typically used without positions by service industries where flexibility and organizational change are key features.

### 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.



## FAQs for Define Workforce Structures

### What's the difference between a job set and a job family?

A job family is a group of jobs that have different but related functions, qualifications, and titles. They are beneficial for reporting. You can define competencies for job families by associating them with model profiles.

A job set is an organizational partition of jobs. For example, a job set can be global and include jobs for use in all business units, or it can be restricted to jobs for a specific country or line of business. When you select a job, for a position or an assignment, the available jobs are those in the set associated with the business unit in which you are working, and also those in the Common set.



---

## Common Applications Configuration: Define Persons for Incentive Compensation

### 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.

## **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.

Attribute	Value
Department	Finance Department
Job	Sales Manager
Assignment Status	Active

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.

Attribute	Value
Manager with Reports	Yes
Assignment Status	Active

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.

Attribute	Value
Department	ABC Department
System Person Type	Employee
Assignment Status	Active

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:

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

## 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.

Attribute	Value
System Person Type	Employee
Assignment Status	Active

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.

Attribute	Value
Business Unit	Sales
Assignment Status	Active
Manager with Reports	Yes

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.

Attribute	Value
System Person Type	Retiree
Assignment Status	Inactive

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.

Attribute	Value
Department	Sales

Job	Sales manager
Grade	6
Assignment Status	Active

In the role mapping, you include the:

- Sales Manager role, and select the **Autoprovision** option
- Sales Associate role, and select the **Requestable** option



---

# Common Applications Configuration: Define Trading Community Details for Incentive Compensation

## Define Source Systems for Incentive Compensation

### 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.

## **Define Geographies**

### **Defining Address Cleansing: Explained**

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



address cleansing validates both the geography attributes and the address line attributes.

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

You can specify the real time address cleansing level for each country by choosing either **None**, meaning that there is no real time address cleansing, or by choosing **Optional**, meaning that you will have the choice to cleanse addresses. Once you have enabled address cleansing for a country a **Verify Address** icon appears at address entry points in the application. You can then click the icon to perform address cleansing and receive a corrected, standardized address. If Trading Community Data Quality does not find a matching address the application will alert you.

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

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

### Geography Structure

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

### Geography Hierarchy

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

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

### Geography Validation

After defining the geography hierarchy, you need to specify the geography validations for the country. You can choose which address style formats you would like to use for the country, and for each selected address style format you can map geography types to address attributes. You can also select which

geography types should be included in geography or tax validation, and which geography types will display in a list of values during address entry in other user interfaces. The geography validation level for the country, such as error or warning, can also be selected.

## Geography Structures: Explained

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

You can use the geography structure to establish:

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

### How Geographies Can Be Related

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

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

---

#### Note

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

---

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

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

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

---

#### Note

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

---

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

# Geography Hierarchy: Explained

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

For example, in the geography hierarchy the state of California is defined as the parent of San Mateo county, which is the parent of Redwood City, which is the parent of the postal code 94065. If you enter just 94065, the application can determine that the postal code is in California, or that the corresponding city is Redwood City.

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

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

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

## Geography

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

## Geography Type

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

## Geography Usage

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

## Master Reference Geography Hierarchy

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

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

### User Defined Zones

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

## Geography Validation: Explained

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

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

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

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

### Map to Attribute

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

---

#### Note

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

---

### Enable List of Values

Once a geography type is mapped to an attribute, then you can specify whether the geography type will appear in a list of values during address entry in user interfaces. It is very important to review carefully if you want to enable a list of

values. You should only enable a list of values if you have sufficient geography data imported or created for that geography. Once you have enabled a list of values for an address attribute, you can only select the geography data available for the geography type. This means that if a specific geography value is not available in the geography hierarchy, you cannot create an address with a different geography value.

## Tax Validation

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

## Geography Validation

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

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

---

### Note

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

---

## Geography Validation Control

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

These are the geography validation levels you can choose:

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

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

---

**Note**

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

---

## Importing Geographies: Explained

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

Consider the following when importing geographies:

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

### File-Based Import Option

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

## Geography Loader Process Option

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

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

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

---

#### Note

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

---

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

File-Based Import Entities	Interface Tables	Destination Tables	Application Object
ImpGeography	HZ_IMP_GEOGRAPHIES	HZ_GEOGRAPHIES HZ_GEOGRAPHY_IDENTITY HZ_GEOGRAPHY_TYPES HZ_HIERARCHY_NODES	Geography

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

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

The following table summarizes the key decisions for this scenario:

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

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

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

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

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

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

### Determine what information is in the source file

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

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



## Create and schedule the import activity

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

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

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

---

### Note

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

---

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

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

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

---

**Note**

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

---

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

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

8. Click **Next**.

### Monitor the import results

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

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

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

## Importing and Exporting Territory Geography Zones: Explained

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

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

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

## Import the master reference geography data

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

## Define your territory geography zones

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

## Export the territory geography zones

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

## Import the territory geography zones

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

---

### Note

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

---

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

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

The following table summarizes the key decisions for this scenario.

Decisions to Consider	In This Example
Copy an existing country structure?	No, create a new country structure.
What is the structure of the geography types?	Create geography types with the following ranking structure:  1. County 2. Post Town

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

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

## Defining the geography structure

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

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

## Defining the geography hierarchy

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

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

8. Click the **Create** button.
9. In the **Create Post Town** page, Primary and Alternate Names section, enter Reading in the **Name** field.
10. Click Save and Close.

## Defining the geography validations

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

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

## FAQs for Define Geographies

### When do I define address cleansing?

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

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

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

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

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

States country geography structure, then the State level cannot be deleted in the country geography structure.

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

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

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

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

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

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

## **Define Import for Incentive Compensation**

### **Trading Community Model Data Import Objects: Explained**

Import objects are business entities that can be imported into the trading community model registry, for example, competitors, partners or resource teams. When you create a data import batch you should choose which business entity, or object, you are importing from the batch into the trading community model registry. For example, if you are responsible for resource management, you might want to import objects such as employee resource and resource team.

The import process flow will change according to which object you have selected. There are two import process flows for the following sets of objects:

1. Customer, reference, competitor, and custom party.
2. Employee resource, resource team, partner, and partner contact.

#### **Customer, Reference, Competitor, and Custom Party**

When you select these objects you will receive the option to check for duplicates within the import batch before the import, and the option to check for duplicates between the import batch and the trading community model registry before import. You will also be able to choose to preview data before it is imported, specify if addresses will be cleansed before import, and set how many errors you will allow before the import is terminated.

#### **Employee Resource, Resource Team, Partner, and Partner Contact**

If you choose to import these objects you will not be able to deduplicate the batch or registry data. However, you will be able to choose to preview data before it is imported, specify if addresses will be cleansed before import, and set how many errors you will allow before the import is terminated.

## Defining the Import Process for Customers and Consumers: Points to Consider

Oracle Fusion Applications provides two utilities to import data, File-Based Import and Data Import. File-based Import allows you to import data directly from a CSV or XML file. Data Import imports data from interface tables once the tables are populated by an external process. Data Import cannot be used in a Cloud deployment, because you cannot populate interface tables using an external process in a Cloud deployment. In Cloud deployments, you can only use File-based import to import data. However, even if you are using File-based Import, you can use the Data Import dashboard to monitor the import job and view reports and statistics. You can choose to pause a File-Import batch, and use Data Import work area to configure data quality services, monitor import jobs, view errors, and view batch statistics. Import activities are paused if the **HZ\_IMP\_PAUSE\_FILE\_IMPORT** profile option is set to Yes in the Manage Import Profile Options page. If the profile option is set to Yes, all submitted import activities are sent for administrator review and appear in the Data Import Batches Overview page.

Before importing data into the registry, you need to decide if you want to use the data quality services and if so, how you want to configure the data quality services.

The data quality services are:

- Batch deduplication
- Registry deduplication
- Import to registry options

### Batch Deduplication

Within the batch deduplication page you can decide if you want to identify and resolve duplicates within the batch that you are importing from the interface tables. If you want to check for duplicates you need to choose what match configuration rule you want to use to identify duplicates for each entity. Then you need to specify what action will be taken on the persons, organizations, and address duplicates found within the batch. Your specified actions will be performed on the batch before the data is imported into the registry.

### Registry Deduplication

Similar to batch deduplication, registry deduplication identifies duplicates between the data in the batch and the data in the registry before the data is imported into the registry. If you want to check for duplicates you need to choose what match configuration rule you want to use to identify duplicates for each entity. Then you need to specify what action will be taken on the persons, organizations, and address duplicates found in the registry deduplication check. Your specified actions will be performed when you import the batch into the registry.

### Import to Registry Options

When defining an import process you can decide whether to run the import process in preview mode, or you can choose to load the data directly into the

registry without previewing the data. You can also choose to cleanse addresses prior to import, validate addresses in an import batch against geography data, and define an error limit for the batch.

### **Import Process Mode**

You can choose to run the import batch in preview mode, or you can skip the preview and load the data directly into the registry.

If you select to run the batch in preview mode you will be able to review information about the level of duplicates or incorrect addresses in the batch data before the data is actually imported. You will also be able to preview how many records will be created and how many records will be updated for each entity. You can then continue to import the batch, or you can amend the match configuration rules and actions to be taken on the identified duplicates and then rerun the batch to review the data again.

If you do not want to review the batch data before it is imported into the registry, then you can choose to skip the preview and allow the data to be loaded into the registry as soon as preprocessing is complete. You may prefer not to preview the batch data if the data source is frequently used.

---

#### **Note**

The Define Import: Import to Registry page is the only place that you can specify if you want to run the batch in preview mode. Once the option to skip the preview mode is selected, and you submit the batch for processing, you will not be able to review the batch data before it is imported.

---

### **Cleanse Addresses**

You can choose to validate the addresses in the interface tables before importing them into the registry. The addresses are validated using an integrated third party service that verifies addresses and corrects them if they are incorrect.

### **Geography Validation**

You can choose to validate the addresses in an import batch against geography data before importing them into registry. The addresses are validated against the master reference geography data, according to the geography-based address validation settings for each country. The addresses with validation errors are not imported. However, Geography Name Referencing is run on all imported addresses regardless of this profile option setting.

### **Error Handling Limit**

You can define how many process errors can be generated by the import batch process before the process terminates automatically. Error reports are generated by the application for you to review.

## **Performing What-If Analysis on Data Import Batches: Worked Example**

This example demonstrates how to perform What-If analysis on a data import batch that has been processed and has completed with a status of pre-import completed. The match configuration is redefined and the import process is



resubmitted. The batch deduplication actions are then amended, and the batch import is completed.

The following table summarizes key decisions for this scenario.

Decisions to Consider	In This Example
Do you want to redefine batch deduplication match configuration?	Yes, a different match configuration is selected for the organizations entity.
Do you want to redefine registry deduplication match configuration?	Yes, a different match configuration is selected for the persons entity.
What actions do you want to take on Persons, Organizations, and Addresses duplicates?	<ul style="list-style-type: none"><li>• Within registry deduplication, choose <b>Do not import duplicate records</b> for Persons and Organizations.</li><li>• Within registry deduplication, choose <b>Import duplicate records</b> for Addresses.</li></ul>

## Prerequisites

1. The data import batch has been created.
2. The data is uploaded into the interface tables.
3. The batch is imported and has completed with a status of pre-import completed.

## Viewing the What-If Analysis

1. On the Data Import Batches Overview page, click on the batch ID URL.
2. On the Edit Data Import Batch page, review the summary and import process performance information. Click **Import Details** to open the What-If analysis page.
3. On the Import Process Details page, click the **Batch Deduplication** tab. Check that you are satisfied with the batch deduplication results.
4. On the Import Process Details page, click the **Registry Deduplication** tab. Check that you are satisfied with the registry deduplication results.
5. On the Import Process Details page, click the **Address Cleansing** tab. Check that you are satisfied with the address cleansing results.

## Redefining the Match Configuration and Resubmitting the Import Process

The results of the batch and registry deduplication are not as expected and so the match configurations need to be redefined.

1. On the Import Process Details page, click **Cancel**.
2. On the Data Import Batches Overview page, click on the batch name. Click **Actions** and then click **Import**.
3. On the Define Import: Batch Deduplication page, choose a different match configuration for the organizations entity. Click **Next**.
4. On the Define Import: Registry Deduplication page, choose a different match configuration for the persons entity. Click **Next**.
5. On the Define Import: Import Registry page, click **Submit**

## Changing the Action for Duplicates within the What-If Analysis

You want to view the What-If analysis for the new match configurations that you selected for the batch.

1. On the Data Import Batches Overview page, click on the batch ID URL.
2. On the Edit Data Import Batch page, review the summary and import process performance information. Click **Import Details** to open the What-If analysis page.
3. The new match configurations have produced satisfactory results, but you would like to change the actions that will be carried out on the duplicates. On the Import Process Details page, click the **Registry Deduplication** tab.
4. For the Persons and Organizations duplicates, choose **Do not import duplicate records** from the choice list.
5. For the possible duplicates for Addresses, choose **Import duplicate records** from the choice list.
6. Click **Complete Import**.

## Incentive Compensation Collection Setup Tasks: Explained

There are prerequisite setup tasks that you must perform in Oracle Fusion Incentive Compensation before importing and collecting the first set of transactions. You must also map the source data columns to the staging table columns.

### Prerequisite Setup Tasks

Entities	Description
Calendars	Before running the Collection process, you must define your calendars and intervals as well as open the period for which you want to collect data.
Parameters	Determine whether to set up and use the incentive compensation Crediting process as well as when you will run the Classification process. Also set up processing and payment currency options.
Currency Conversions	You can credit, calculate, and pay in either the participant's home currency or process using a single currency within each incentive compensation business unit. Even if you process in a single currency (operating currency), you may still pay using each participant's home currency. To process transactions without error, you must define currency conversion rates in the Application Setup work area. Also, you must associate home currency to each participant. If using operating currency for processing and payment, then default the participant's home currency to <b>Operating Currency</b> during participant import.
Earning Types	Define any unit of measure that you use for processing, other than monetary amount (already defined).

Flexfields	There are 150 additional transaction entity attributes for use in processing. Configure these descriptive flexfields for use as your business dictates. For example, if you want to include Customer on the transaction, then define one of the flexfields for this, as well as point to a data source (if other than Oracle Fusion Trading Community Architecture) to provide lookups to use for validation.
Tables and Columns	Enable any additional attributes as well as disable relevant attributes used in the Crediting or Classification processes as well as the calculation expression builder.

Use the Oracle Data Integrator Mapping screens to set up the transaction sources and mappings between various columns to the Oracle Fusion Incentive Compensation staging table (CN\_TP\_TRANSACTIONS\_STAGING\_T). This table has the same schema as the transaction table (CN\_TP\_TRANSACTIONS\_ALL), including the column called CHANGED\_TRX\_FLAG, which distinguishes whether a transaction is new or updated. You are responsible for maintaining the logic for importing new or updated source application transactions into the staging table using Oracle Data Integrator or another extraction, transformation, and load tool. Track transactions adjusted in the source application using the change data capture mechanism supported by Oracle Data Integrator, or any other custom logic. In staging, the application compares the composite key transaction number and transaction type to determine if the transaction is present before performing the Obsolete process.

---

#### Restriction

You cannot customize the Oracle Data Integrator mappings that are delivered with the product and used to load the data from the staging table into the transaction table. You can create your own custom script to use instead.

---

## FAQs for Define Import

### Can I redefine the data import process for an already imported batch and reimport it?

Yes. If the data is still available in the interface tables and the batch status is **Preimport Completed**, **Completed with Errors**, **Error**, or **Terminated when Error Limit Reached**, then you can redefine the data import process and reimport the batch. However, once a batch has been successfully imported then you will not be able to reimport the batch, even if the data is present in the interface tables.

### How can I view the errors that occurred during preimport processing?

You can view any errors that occurred after submitting the batch for import by selecting the batch in the data import batches Overview page, and then click **Report**.



---

# Common Applications Configuration: Define Resources for Incentive Compensation

## Identifying Resources: Explained

The Identify Resources step in the Manage Resources task is only needed to identify an existing employee, contingent worker, or partner member as a resource. Usually they are identified as resource in the Manage Users task, or in the Partner Center. If you have created partner members or internal users in the system without making them resources, you can identify them as resources in the Identify Resources step. Until you identify employees, contingent workers, and partner members as resources, you cannot assign them work objects.

---

### Note

Resources need not necessarily belong to an organization, nor do they need to have specific roles assigned. However, it is best to always associate resources with an organization either as managers or as members. Similarly resources should also have at least one role as part of their organization membership. When you identify users as resources, all you indicate is that these new resources can now be assigned work within the deploying company.

---

## Resource Directory: Explained

The Resource Directory offers detailed information about all the resources within the deploying organization. The Resource Directory also enables you to find and communicate with other resources, and to network and collaborate with them.

Use the Resource Directory to perform the following tasks:

- View and modify your profile
- View your organization and team membership information
- View information related to other organizations and teams
- View the profiles of other resources
- Communicate with other resources

## Resources: How They Work within a Team

You can include resources from different resource organizations to work together on a work object as members of the same resource team. You can also include entire resource organizations into a resource team. Generally what resources can do is controlled by their resource organization membership and their hierarchy. Resource teams provide a flexible way of bringing resources together without any organizational or hierarchy-based restrictions.

### Assigning Resources to Teams

You can assign identified resources to teams and assign them roles within the team. Each resource can have a specific role within a team. Thus, a resource may play different roles in different teams.

## Team Usage: Explained

You use teams to ensure optimal utilization of existing resources across the deploying company without changing their organizational membership and hierarchy. You can also use resource teams as a quick reference to a group of related resources. This simplifies resource assignment, and you can quickly assign an entire team to tasks in a single move. Team usage indicates the usage of a resource team. This simplifies further task assignment, ensuring that teams are assigned tasks that are associated with or related to their listed usages.

## Provisioning Security Roles: Explained

Security role provisioning is the process of automating the provisioning and de-provisioning of security roles based on resource role assignment to resources. Once security roles are provisioned to resources, they can access the tasks and data enabled for the security role.

### Difference between Resource Roles and Security Roles

Resource roles indicate who a person is to the deploying company. As such, resource roles are used for filtering resources and for generating reporting hierarchies in addition to being used to define security policies. A key difference between a security role and resource role is that a resource role may be assigned to a resource without a user account, while a security role can only be provisioned to a resource who has a user account. So while in some cases the resource role may be defined at the same granularity as the security role and used to automate security role provisioning, the resource role concept remains separate from security roles.

## FAQs for Define Resources

### What's the difference between a partner resource and an internal resource?

The main difference between an internal resource and a partner resource is the company for whom each works. While the internal resource is an employee

or contingent worker of the deploying company, the partner resource is an employee of the partner company.

The methodology used to create resources of these two types is also different. While the partner administrator or channel manager creates a new partner resource through the Oracle Fusion Partner Management applications, internal resources are added using the Manage Users, Hire Employee, or Import Person and Organization task.

Another difference between partner resources and internal resources is that partner resources cannot access the Resource Directory while internal resources can.

### **Can I create an employee or contingent worker resource?**

No. You can only identify existing employees and contingent workers as resources in the Manage Resources task, but you cannot create a new employee or contingent worker in the Manage Resources task.

You can create an employee or contingent worker using Manage Users task, Hire Employee task, or Import Person and Organization task.





---

# Common Applications Configuration: Define Security for Incentive Compensation

## 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)

---

Security setup and administrative tasks performed by product administrators and implementation consultants, such as managing HCM security profiles, are presented in the documentation for those products.

### 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

- You manage duties, security policies, and data role templates in the Authorization Policy Manager. Refer to the Oracle Fusion Middleware Authorization Policy Manager Administrator's Guide (Oracle Fusion Applications Edition).

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.
- For a detailed functional explanation of the Oracle Fusion Applications security approach, refer to the following guides.

See: Oracle Fusion Applications Security Guide

See: Oracle Fusion Applications Security Hardening 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

(OIM). Authorization policy management involves managing duty roles, data role templates, and data security policies. Refer to the Oracle Fusion Middleware Authorization Policy Manager Administrator's Guide (Oracle Fusion Applications Edition).

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

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.

# 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

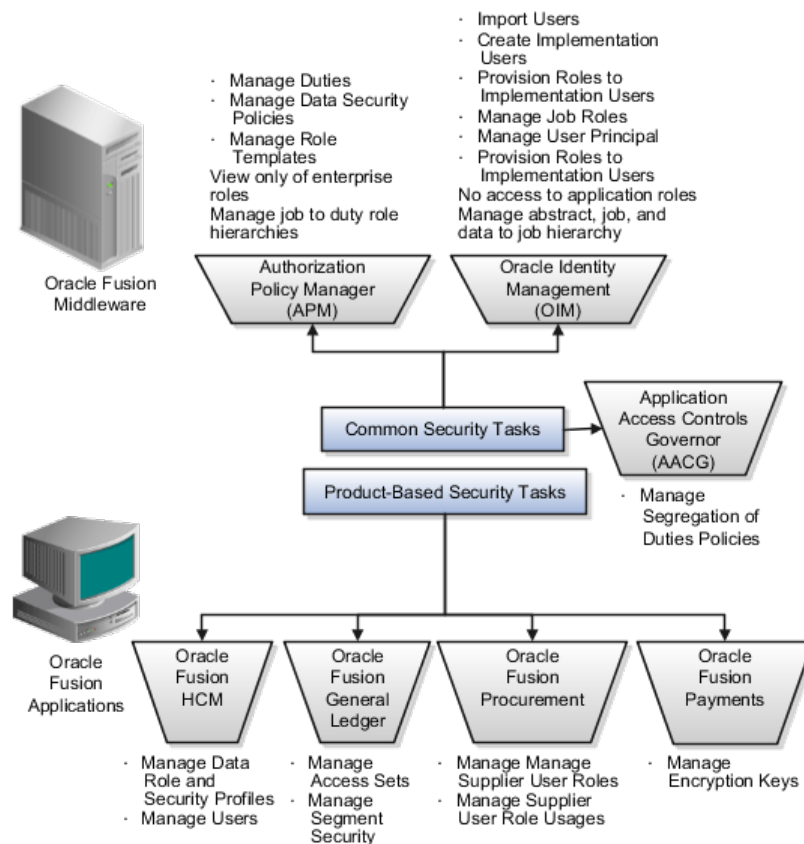
As a security guideline, provision a dedicated security professional with the IT Security Manager role at the beginning of an implementation, and revoke that role from users provisioned with the Application Implementation Consultant role.

---

### 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.

Example Task	Oracle BPM Task	Supporting Tools	Details
View duty roles inherited by a job role	Manage Duties	<ul style="list-style-type: none"> <li>Authorization Policy Manager (APM)</li> </ul>	Each logical partition or pillar contains a collection of application roles, and function and data security policies.
View entitlement or policies carried by a job role	Manage Duties	<ul style="list-style-type: none"> <li>APM</li> </ul>	In LDAP, the policy store stores application roles and the identity store stores enterprise roles.
Add a job role to a role hierarchy	Manage Job Roles	<ul style="list-style-type: none"> <li>Oracle Identity Management (OIM)</li> </ul>	The identity store in LDAP stores enterprise roles.
Add a duty role to a role hierarchy	Manage Duties	<ul style="list-style-type: none"> <li>APM</li> </ul>	LDAP stores the role hierarchy and the spanning of roles across multiple pillars or logical partitions.
Create a hierarchy of enterprise (abstract, job, data) roles	Manage Job Roles	<ul style="list-style-type: none"> <li>OIM</li> </ul>	
Create a hierarchy of (application) duty roles	Manage Duties	<ul style="list-style-type: none"> <li>APM</li> </ul>	
Create a new job role	Manage Job Roles	<ul style="list-style-type: none"> <li>OIM</li> </ul>	The identity store in LDAP stores enterprise roles.
Change duty roles inherited by a job or abstract role	Manage Duties	<ul style="list-style-type: none"> <li>APM</li> </ul>	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.
Create a new duty role	Manage Duties	<ul style="list-style-type: none"> <li>APM</li> </ul>	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.

View Segregation of Duties (SOD) policies respected by a duty role	Manage Application Access Controls	<ul style="list-style-type: none"> <li>Application Access Controls Governor (AACG) in Governance, Risk, and Compliance Controls (GRCC)</li> </ul>	The Security Reference Manuals (SRM) document the segregation of duties (SOD) policies respected within each job role
View SOD policy violations carried by the duty roles inherited by a job role	Manage Application Access Controls	<ul style="list-style-type: none"> <li>AACG in GRCC</li> </ul>	The Security Reference Manuals (SRM) document the SOD policies respected within each job role
View SOD policy violations	Manage Segregation of Duties Policies	<ul style="list-style-type: none"> <li>AACG in GRC</li> </ul>	The SRM documents the SOD conflicts for each job role
View the data security policies carried by a job, abstract, and data roles	Manage Data Security Policies	<ul style="list-style-type: none"> <li>APM</li> </ul>	<p>Oracle Fusion Data Security stores data security policies in the policy store.</p> <p>Data security can also be defined in application pages provided by Oracle Middleware Extensions for Applications (FND)</p>
Create and update HCM security profiles	Manage Data Role and Security Profiles	<ul style="list-style-type: none"> <li>Oracle Fusion HCM</li> </ul>	This task does not include assigning data roles to the users, which is supported by user provisioning tasks.
Create (generate) a data role	<ol style="list-style-type: none"> <li>Manage Role Templates</li> <li>Manage Data Roles and Security Profiles</li> </ol>	<ul style="list-style-type: none"> <li>APM</li> <li>Oracle Fusion HCM</li> </ul>	<p>Data roles are generated automatically based on data role templates and enterprise setup. Changes to data role templates generate new or changed data roles.</p> <p>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	<ul style="list-style-type: none"> <li>APM</li> </ul>	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	<ul style="list-style-type: none"> <li>APM</li> </ul>	
Create or edit an existing data role template	Manage Role Templates	<ul style="list-style-type: none"> <li>APM</li> </ul>	

Secure common objects such as attachment categories or profile options	Manage Data Security Policies	<ul style="list-style-type: none"> <li>• APM</li> </ul>	Data security can also be defined in application pages provided by Oracle Middleware Extensions for Applications (FND)
View, create, update encryption keys used to secure attributes of personally identifiable information	Manage Encryption Keys	<ul style="list-style-type: none"> <li>• Oracle Fusion Payments</li> </ul>	
View, create, update Data Access Sets used to secure Ledgers and Ledger Sets	Manage Data Access Sets	<ul style="list-style-type: none"> <li>• Oracle Fusion General Ledger</li> </ul>	
View, create, update accounting flexfield segment security rules	Manage Security Segments	<ul style="list-style-type: none"> <li>• Oracle Fusion General Ledger</li> </ul>	
View or update the set of job roles that can be provisioned to supplier users	Manage Supplier User Role	<ul style="list-style-type: none"> <li>• Supplier Portal</li> <li>• Sourcing</li> </ul>	These tools are in the Oracle Fusion Procurement product family
Determine the supplier job roles that the supplier self service administrator can provision to supplier users	Manage Supplier User Role Usages	<ul style="list-style-type: none"> <li>• Supplier Portal</li> <li>• Sourcing</li> </ul>	These tools are in the Oracle Fusion Procurement product family
Set default supplier job roles based on the set of supplier roles that are defined by performing the Manage Supplier User Roles task	Manage Supplier User Role Usages	<ul style="list-style-type: none"> <li>• Supplier Portal</li> <li>• Sourcing</li> </ul>	These tools are in the Oracle Fusion Procurement product family
Create a new implementation user	Create Implementation Users	<ul style="list-style-type: none"> <li>• OIM</li> </ul>	
Import legacy users	<ul style="list-style-type: none"> <li>• Import Worker Users</li> <li>• Import Partner Users</li> </ul>	<ul style="list-style-type: none"> <li>• OIM</li> </ul>	
Create a new user	Manage Users	<ul style="list-style-type: none"> <li>• HCM</li> </ul>	<p>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.</p> <p>Creating a new user automatically triggers role provisioning requests based on role provisioning rules.</p>

Provision roles to a user	<ol style="list-style-type: none"> <li>1. Provision Roles to Implementation Users</li> <li>2. Manage Users</li> </ol>	<ul style="list-style-type: none"> <li>• OIM</li> <li>• Oracle Fusion HCM</li> <li>• Oracle Fusion CRM</li> <li>• Oracle Fusion Suppliers</li> </ul>	<p>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.</p> <p>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.</p> <p>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.</p>
View the job, abstract, and data roles provisioned to a user	<ol style="list-style-type: none"> <li>1. Manage Users</li> <li>2. Manage User Principal</li> <li>3. Provision Roles to Implementation Users</li> </ol>	<ul style="list-style-type: none"> <li>• Human Capital Management (HCM)</li> <li>• OIM</li> </ul>	<p>LDAP stores users, roles and provisioning information.</p> <p>The Manage User Principal and Provision Roles to Implementation Users tasks are not needed once implementation is complete.</p>
Revoke role from user.	Manage Users	<ul style="list-style-type: none"> <li>• HCM</li> </ul>	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	<ul style="list-style-type: none"> <li>• OIM</li> </ul>	
View audit logs	Not applicable	<ul style="list-style-type: none"> <li>• Oracle Enterprise Manager</li> </ul>	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.

Data security feature	Does what?
Data security policy	Grants access to roles by means of entitlement
Role	Applies data security policies with conditions to users through role provisioning.

Data role template	Defines the data roles generated based on enterprise setup of data dimensions such as business unit.
HCM security profile	Defines data security conditions on instances of object types such as person records, positions, and document types without requiring users to enter SQL code
Masking	Hides private data on non-production database instances
Encryption	Scrambles data to prevent users without decryption authorization from reading secured data

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.

Who	can do	what	on which set of data
Accounts payable managers	view	AP disbursements	for their business unit

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.

---

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 security policies (the entitlement is Manage Data Security Policy). This entitlement provides the access necessary to perform the Manage Data Security Policies task in APM.

## 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.

Method of securing data	Reason	Example
Data roles apply explicit data security policies on job and abstract roles	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.	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.
Data security policies	Define data access for application roles and provide inheriting job and abstract roles with implicit data security	Projects

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, `REGION='WEST'` is a static object instance set and `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.

---

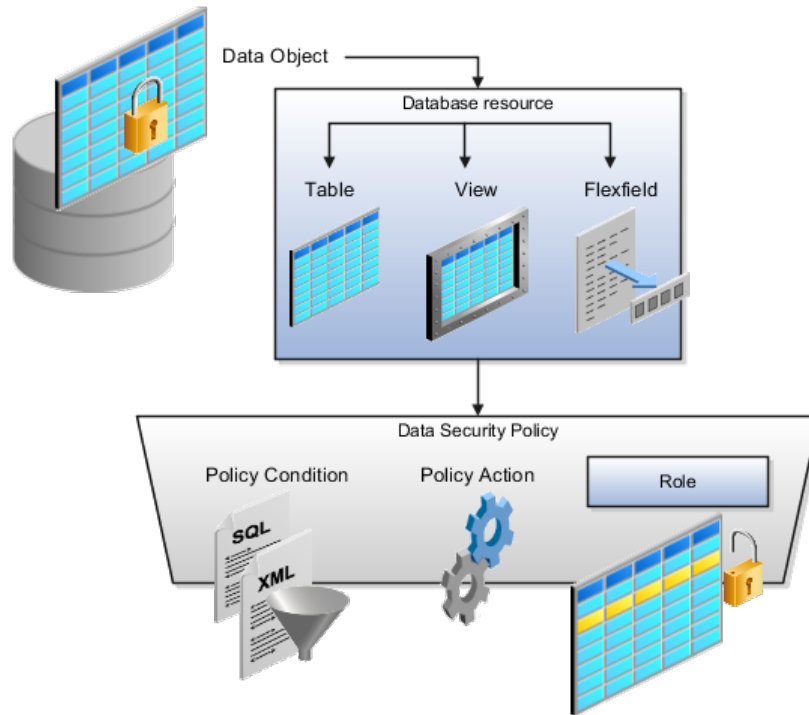
## Manage Data Security Policies

### 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 an 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

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 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.

Attribute	Value
Department	Finance Department
Job	Sales Manager
Assignment Status	Active

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.

Attribute	Value
Manager with Reports	Yes
Assignment Status	Active

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.

Attribute	Value
Department	ABC Department
System Person Type	Employee
Assignment Status	Active

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:

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

### 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.

Attribute	Value
System Person Type	Employee
Assignment Status	Active

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.

Attribute	Value
Business Unit	Sales
Assignment Status	Active
Manager with Reports	Yes

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.

Attribute	Value
System Person Type	Retiree
Assignment Status	Inactive

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.

Attribute	Value
Department	Sales

Job	Sales manager
Grade	6
Assignment Status	Active

In the role mapping, you include the:

- Sales Manager role, and select the **Autoprovision** 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.

---

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

### Prerequisites

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

Attribute	Value
System Person Type	Employee
Assignment Status	Active

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:

Field	Value
Last Name	Williams
First Name	Gail
E-Mail	gail.williams@vision.com
Hire Date	4/12/11

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:

Field	Value
Legal Employer	Vision Corporation
Business Unit	Vision Canada
Department	Human Resources

## 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.

## FAQs for Manage Users

### What happens if I send the user name and password?

An e-mail containing the user name and password is sent to the user's primary work e-mail address. If the user has no primary work-email address, then the user name and password are sent to the primary work e-mail address of the user's line manager, if available; otherwise, no notification is sent.

You can select **Send user name and password** only if these details have not already been sent for this user: the user name and password can be sent once only for any user. If this option is available for selection but you do not select it, then you can run the process Send User Name and Password E-Mail Notifications later to notify users of their user names and passwords.

---

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

## 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

enforced in implementation, test, and production instances of Oracle Fusion Applications.

## 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 Manager 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 Manager 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.

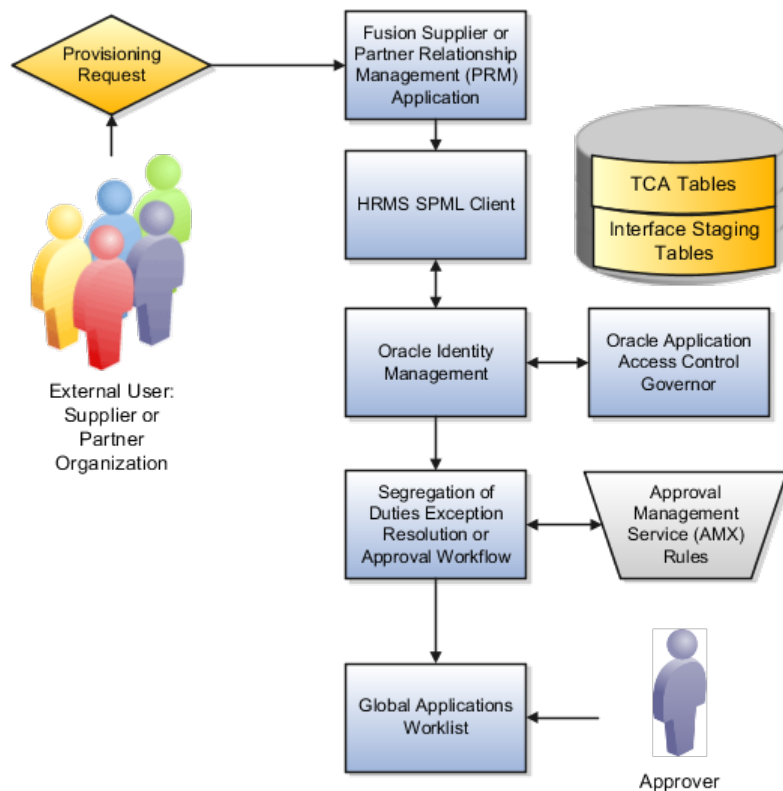
---

### 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.



## Tables

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.

Status	Conflicts	Current State
SOD_CHECK_IN_PROGRESS	Unknown	Request sent to AACG and waiting for response
SOD_REMEDIATION_IN_PROGRESS	Conflict found	AACG detected violations and remediation is in progress
SOD_CHECK_APPROVED	No conflict found	No SOD violations found
SOD_CHECK_REJECTED	Conflict found	AACG detected violations that cannot be remediated
SOD_REMEDIATION_APPROVED	Conflict found	AACG detected violations that are approved

SOD_REMEDIATION_REJECTED	Conflict found	AACG detected violations that are rejected by approver
--------------------------	----------------	--

In the absence of an SOD exception, OIM provisions all relevant users.

---

#### 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\_REMEDIATION\_APPROVED status to the SPML client.

If the approver rejects the request, OIM sends an SOD\_REMEDIATION\_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 Approval Management for Incentive Compensation

## 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

- Approval management is an extension of the human workflow services of Oracle SOA Suite. Refer to the Oracle Fusion Middleware Modeling and Implementation Guide for Oracle Business Process Management.

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

- Task configuration involves managing policies that control approval flows. Refer to the Oracle Fusion Middleware Modeling and Implementation Guide for Oracle Business Process Management.

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

## Approval Management Configuration Options for Oracle Fusion Incentive Compensation: Explained

The approval management extensions to the human workflow services of Oracle SOA Suite have the following default configuration options for Oracle Fusion Incentive Compensation.

Configuration Option	Default Value	Effect of Default Value
Ad hoc insertion of approvers	True	Allow ad hoc insertion of approvers in the approval list. Users who add approvers may also modify or remove the approvers that they add.
Allow delegate	True	Allow approvers to delegate their approval responsibilities to other users. One approver replaces another, but the approver list is otherwise unaltered.
Allow pushback	True	An approver can push the transaction back to the previous approver, who thereby has a second opportunity to review the transaction.
Allow reassign	True	Any approver can reassign the approval to a different approver. The approval list is recalculated based on the new approver.
Allow request information	True	Approvers can request more information from another approver or the person who submitted the transaction.
Allow self-approval	False	The person who submits the transaction cannot approve it.

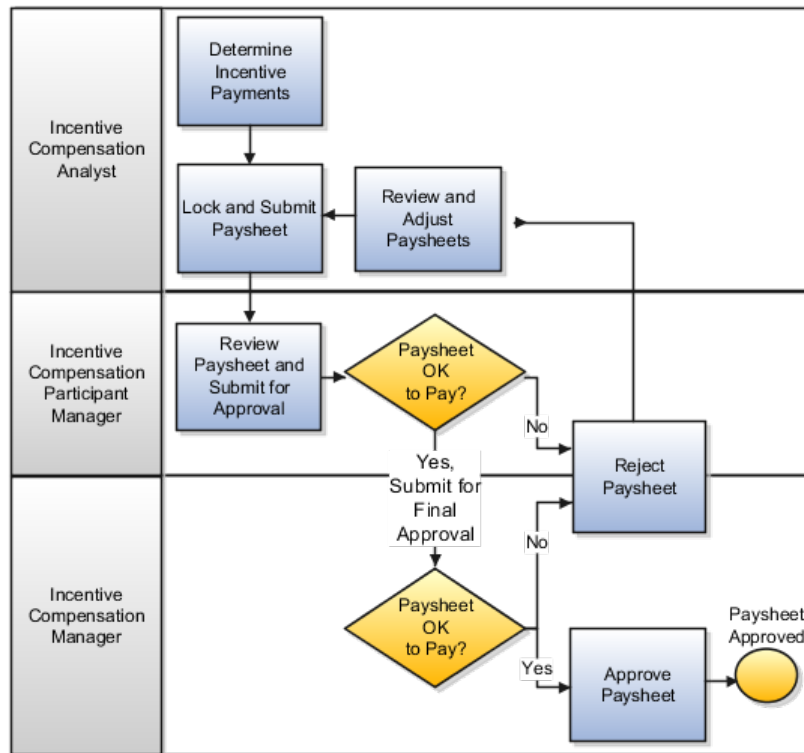


Allow withdraw	True	The requester or an administrator can withdraw a transaction while the approval process is incomplete. Approvers who have already approved are notified of the withdrawal. The transaction is removed from the work lists of approvers who have not yet approved.
On error notify	Incentive Compensation Application Administrator	An incentive compensation application administrator is notified automatically when an error occurs.
Period before task expires	None	Approval tasks do not expire.
Period before task escalates	None	Approval tasks are not escalated to other approvers.
Escalated approver	None	Approval tasks are not escalated to other approvers.
Repeated approver frequency	Once per approval	An approver receives one notification per transaction, even when the approver appears multiple times in the approver list.
Re-evaluate approver list	True	The approver list is regenerated after every response.
Rejection outcome	Stop all	When an approver rejects a transaction, the approval process stops and the transaction is canceled.

## Incentive Compensation Payment Approval Process: How It Is Configured

This topic covers the incentive compensation Payment Approval business process activities associated with the paysheet statuses, including when and what types of actionable and informational notifications the application sends to incentive compensation (IC) managers and analysts as well as participants and participant managers.

The following figure shows the payment approval business process activities and the job roles that perform them.



## Settings That Affect the Payment Approval Process

The application sends notifications to participants, participant managers, compensation analysts, and compensation managers based on the paysheet statuses and rules defined within the approval management extensions to the human workflow services of Oracle SOA Suite, as well as the analyst payment approval hierarchy in the Participant Assignments work area, on the Manage Analyst Payment Approval Hierarchy page. When you act on a notification, and edit the paysheet status, the business rules send a new notification. Define, add, edit, and delete steps and statuses within the approval management extensions, as required.

### Tip

If you do not require the payment approval business process, in the Define Business Unit Configuration for Incentive Compensation task list, Manage Parameters task, for the **Default Paysheet Approval Status** parameter, select **Approved**, which is the final status before paying the payment batch.

### Rule-based Exceptions using Minimum and Maximum Payment Autoapproval Amount Parameters

The provided payment approval business process includes an exception rule that causes the application to send actionable notifications to compensation managers if the paysheet amount is greater than or equal to X and less than or equal to Y (the application also automatically sets the status to Approved). If the minimum and maximum payment autoapproval amount parameters in the Manage Parameters task are NULL, then the application does not automatically approve any of the paysheets. The application still uses the regular approval business process that is in place and continues the regular approval business process activities for all paysheets.

## How the Payment Approval Process Is Configured

The following table describes the provided approval business process activities, paysheet statuses, who sets the status, and the resulting application actions.

Activity	Status	Status Set By	Application Action
Create paysheets	Unpaid*	Application	Send no notifications.
Create paysheets	Approved	Application	Check the minimum and maximum payment approval amount parameters and if the paysheet amount is between the two values, it sets the status and sends informational notifications to the participant and IC managers.
Request more information from participant, participant manager, or Sales Operations	Request More Information	IC manager, analyst, participants, or participant manager	Send an informational notification to the requestor and a request notification to the person being asked for more information.
Submit locked paysheets	Submitted	IC analyst	Send informational notifications to the relevant participant and analyst, as well as an actionable notification to the relevant IC manager.
Approve paysheets	Approved	IC manager	Send an actionable notification of approval to the relevant analyst and informational notifications of to the relevant participant and participant manager.
Reject paysheets	Rejected	IC manager	Send an actionable notification of rejection to the relevant analyst and informational notifications of to the relevant participant and participant manager.
Reassign paysheet approval	No change	IC manager	Send informational notifications to the analysts associated with the paysheet as well as the requesting incentive compensation manager. Also send an actionable notification to the new IC manager.

\* Based on the **Default Paysheet Approval Status** parameter

## Dispute Management and Approval: How It Is Configured

There are times when participants, their managers, compensation analysts, and compensation managers must create, track, and resolve disputes as part of the incentive compensation process. This effective dispute management includes automated notification to all involved parties.

### How Dispute Management and Approval Is Configured

Create: Complete the required fields as well as all fields relevant to expedite the resolution of your dispute. After you submit the new dispute, the application:

- Creates the dispute record
- Sets the task outcome to **New**
- Assigns the dispute to the compensation analyst associated with the participant using the approval workflow

If you are a compensation analyst or manager and are creating the dispute for a participant assigned to you, the application automatically assigns the dispute to you, the initiator. If you are the participant or participant's manager, the application uses the approval workflow to identify the compensation analyst associated with you (the participant) or your direct report, the dispute participant.

- Sets the status to **Assigned** when the approval workflow finds the compensation analyst associated with the dispute participant; otherwise, sets the status to **New**
- Sends an actionable notification to the compensation analyst of the new dispute
- Sends an informational notification to the dispute participant

Reassign and delegate: Compensation analysts can reassign one of more of their disputes to other analysts to balance the workload. Compensation managers can delegate disputes among analysts. In both situations, the application:

- Supports the reassignment of disputes to users that are part of the Oracle Fusion Human Capital Management (HCM) hierarchy
- Assigns the dispute to the selected (new) compensation analyst
- Sets the status to **Assigned**
- Sends an actionable notification to the new compensation analyst

Request more information: The compensation manager or analyst requests more information from the participant, participant's manager, or Sales Operations team--any incentive compensation user. The application:

- Sets the status to **Information requested**
- Sends an actionable notification to the identified user

Update Information: When the participant, participant's manager, or Sales Operations user update the dispute with the requested information, the application:

- Sets the status to **Assigned**
- Sends an actionable notification to the user who requested the additional information

Approve and reject: The compensation manager or analyst views the details of the dispute. If no further information is required, the analyst or manager approves or rejects the dispute. The application:

- Sets the task outcome to **Approved** or **Rejected**
- Sets the status to **Approved** or **Rejected**
- Sends an informational notification to the dispute initiator, which can be a participant, the participant's manager, and the analyst

Escalate: When the resolution was not as expected, for example, the dispute was rejected, the compensation analyst can escalate the dispute, even a dispute that was closed. The application:

- Sets the status to **Escalated**
- Sends actionable notifications to the compensation analyst assigned to the dispute as well as all of the users with the Compensation Manager role
- Sends an informational notification to the participant's manager, if the participant is the one escalating the dispute

Withdrawn: The dispute creator can withdraw the dispute before it is approved or rejected. The application:

- Sets the status to **Withdrawn**

## Manage Task Configurations for Incentive Compensation

### Enabling an Incentive Compensation Attribute as a Qualifier for Classification or Crediting Rules: Worked Example

This example demonstrates how to create a custom incentive compensation attribute with a list of values for use in crediting rules for those situations where the delivered attributes do not fulfill business requirements. The application has these attributes (in the CN\_TP\_TRANSACTIONS\_ALL table) enabled for use in crediting and classification rules:

- AREA\_CODE
- CITY
- COUNTRY
- CUSTOMER\_ID

---

#### Tip

Use the customer (directory id) or customer name range for matching.

- INVENTORY\_ITEM\_ID

---

#### Tip

Use the product id or sales catalog hierarchy for matching.

- ORG\_ID
- PARTICIPANT\_ID
- POSTAL\_CODE
- PROVINCE
- ROLE\_ID (enabled for crediting rules only)
- SALES\_CHANNEL
- SOURCE\_ORG\_ID
- STATE
- TERR\_NAME
- TRANSACTION\_AMT\_FUNC\_CURR
- TRANSACTION\_QTY
- TRANSACTION\_TYPE

The three tasks covered in this topic are:

- Run SQL script to enable a custom attribute as a qualifier for crediting rules.
- Enable the custom attribute.
- Edit and deploy the relevant descriptive flexfield.

---

#### Tip

Also use the steps in these tasks to create a custom attribute for classification rules by changing the usage\_id from -1001 to -1002.

---

## Running Script to Enable Custom Attribute

In this task, use SQL and a tool such as SQL Developer to create new FND LOOKUP values to use in the choice list on the user interface (optional), run the CN\_RS\_CUST\_ATTR.CREATE\_QUAL application programming interface (API) to create the custom attribute, and update the BATCH\_TOTAL\_SQL to assign the new custom attribute.

1. Create new FND LOOKUP values to use in the choice list on the user interface (optional).

```
-- Example: create a Line of Business lookup
-- LOOKUP_TYPE = CN_LINE_OF_BUSINESS
DECLARE
    l_retcode VARCHAR2(30);
    l_errbuf VARCHAR2(10000);
BEGIN
    -- create the FND lookup TYPE
    fnd_lookup_types_pkg.CREATE_OR_UPDATE_ROW(
        X_VIEW_APPSNAME => 'FND',
        X_LOOKUP_TYPE => 'CN_LINE_OF_BUSINESS',
        X_APPLICATION_SHORT_NAME => 'CN',
        X_MEANING => 'Line of Business',
        X_DESCRIPTION => 'Fusion Example: Line of Business',
        X_REFERENCE_GROUP_NAME => '');
    -- populate the FND lookup values
    fnd_lookup_values_pkg.CREATE_OR_UPDATE_ROW(
        X_LOOKUP_TYPE => 'CN_LINE_OF_BUSINESS',
        X_VIEW_APPSNAME => 'FND',
        X_LOOKUP_CODE => 'COMMERCIAL_LEASE',
        X_MEANING => 'Commercial Leasing',
        X_SET_CODE => '',
        X_DESCRIPTION => 'Commercial Leasing',
        X_ENABLED_FLAG => 'Y',
        X_START_DATE_ACTIVE => sysdate,
        X_END_DATE_ACTIVE => '');
    fnd_lookup_values_pkg.CREATE_OR_UPDATE_ROW(
        X_LOOKUP_TYPE => 'CN_LINE_OF_BUSINESS',
        X_VIEW_APPSNAME => 'FND',
        X_LOOKUP_CODE => 'RESIDENTIAL',
        X_MEANING => 'Residential',
        X_SET_CODE => '',
        X_DESCRIPTION => 'Residential',
        X_ENABLED_FLAG => 'Y',
        X_START_DATE_ACTIVE => sysdate,
        X_END_DATE_ACTIVE => '');
    fnd_lookup_values_pkg.CREATE_OR_UPDATE_ROW(
        X_LOOKUP_TYPE => 'CN_LINE_OF_BUSINESS',
        X_VIEW_APPSNAME => 'FND',
        X_LOOKUP_CODE => 'CONSUMER_LOAN',
        X_MEANING => 'Consumer Personal Loans',
        X_SET_CODE => '',
        X_DESCRIPTION => 'Consumer Personal Loans',
        X_ENABLED_FLAG => 'Y',
```

```

X_START_DATE_ACTIVE => sysdate,
X_END_DATE_ACTIVE => '');
END;
/

```

2. Run the CN\_RS\_CUST\_ATTR.CREATE\_QUAL application programming interface (API) to create the custom attribute.

API Parameter	Description	Sample Value
p_attribute_id	Get the current minimum attribut_id and decrement it by 1  sql> select min(attribute_id) -1 from cn_rs_attributes_all	-2026
p_object_version_number	Not available	1
p_org_id	sql> select organization_id, name from hr_all_organization_u where name like 'Vision Services%' ;	458
p_source_lang	Not available	'US'
p_language	Not available	'US'
p_name	This is the user interface display name and you can enter any name	'Line of Business'
p_description	Not available	'Custom Attribute : Line of Business'
p_attr_usage_id	Use any number or pick the next number from here:  sql> select min(attr_usage_id) from cn_rs_attr_usages_all WHERE usage_id = -1001  It is good to use the same value p_attribute_id, unless it is already assigned.	-2026
p_usage_id	usage_id = -1001 for crediting  usage_id = -1002 for classification	-1001

p_indexed_col_name	<p>Corresponds to cn_rs_usage_details_all table batch_total_sql. Use the next available value for the same data type, in the unused column. To see which indexed_col_name are used:</p> <pre>sql&gt; select     enabled_flag,     batch_total_sql from     CN_RS_USAGE_DETAILS_A where usage_id =     -1001 and org_id = 458 ;</pre>	'SQUAL_CHAR20'
p_display_type	<p>'CHAR' = varchar or character</p> <p>'NUMERIC' = number</p>	'CHAR'
p_source_column_name	<p>From the batch_total_sql field, there are two tables CN_TP_TRANSACTIONS and HZ_PARTIES. Use one of these fields as the source_column_name. If you want to use a field from other tables, then batch_total_sql must include those tables.</p> <pre>sql&gt; desc     fusion.CN_TP_TRANSACT sql&gt; desc     fusion.HZ_PARTIES</pre>	'ATTRIBUTE38'



p_attr_relation_factor	<p>Get the next prime number, see <a href="http://primes.utm.edu/lists/small/1000.txt">http://primes.utm.edu/lists/small/1000.txt</a>. To see which prime number is already used:</p> <pre>sql&gt; select   attr_relation_factor,   indexed_col_name,   attribute_id from   CN_RS_ATTR_USAGES_ALL where usage_id =   -1001 and org_id = 458 order by   attr_relation_factor desc ;</pre>	227
p_comparison_operator_cn	<p>Define the value for this custom attribute. Use the positive value of p_attribute_id.</p> <p>Naming convention: q&lt;p_attribute_id&gt;_cp</p>	'q2026_cp'
p_low_value_char_cn	<p>Populate if using the <b>equal</b> or <b>between</b> parameter.</p> <p>Naming convention: q&lt;p_attribute_id&gt;_low_va</p>	'q2026_low_value_char'
p_high_value_char_cn	<p>Populate if using the <b>between</b> parameter.</p> <p>Naming convention: q&lt;p_attr_usage_id&gt;_high</p>	'q2026_high_value_char'
p_equal_flag	Enable the EQUAL operator	'Y'
p_batch_op_eol	SQL syntax for EQUAL operand if p_equal_flag = 'Y'	'( A.squal_char20 = B.q2026_low_value_char AND B.q2026_cp = "=" )'
p_like_flag	Enable the LIKE operator	'Y'
p_batch_op_like	SQL syntax for EQUAL operand if P_LIKE_FLAG = 'Y'	'( A.squal_char20 like B.q2026_low_value_char AND B.q2026_cp = "LIKE" )'
p_between_flag	Enable the BETWEEN operator	'Y'

p_batch_op_between	SQL syntax for EQUAL operand if P_BETWEEN_FLAG = 'Y'	'( A.squal_char20 between B.q2026_low_value_char and B.q2026_high_value_char AND B.q2026_cp = "BETWEEN" )'
p_convert_to_id_flag	Convert the varchar2 column to id column, set this value to 'Y'	'N'
p_low_value_char_id_cn	Convert the varchar2 column to id column. Only edit if this P_CONVERT_TO_ID_FLAG = 'Y'	null
p_html_lov_sql1	If you want an list of value (LOV) to be associated to the attribute.	'select /*+ FIRST_ROWS(10) */ 1 lookup_id, meaning meaning, lookup_code lookup_code from cn_lookups where lookup_type = "CN_EXAMPLE_LINE_OF_BUSINESS" ,

```
-- Example: ATTRIBUTE38 is enabled for crediting with list of value
-- from CN_LINE_OF_BUSINESS FND lookup
```

```
DECLARE
```

```
l_retcode VARCHAR2(30);
```

```
l_errbuf VARCHAR2(10000);
```

```
BEGIN
```

```
CN_RS_CUST_ATTR.CREATE_QUAL(
```

```
  p_attribute_id => -2026,
```

```
  p_object_version_number => 1,
```

```
  p_org_id => 458,
```

```
  p_SOURCE_LANG => 'US',
```

```
  p_LANGUAGE => 'US',
```

```
  p_name => 'Line of Business',
```

```
  p_description => 'Custom Attribute : Line of Business',
```

```
  p_ATTR_USAGE_ID => -2026,
```

```
  p_USAGE_ID => -1001,
```

```
  p_INDEXED_COL_NAME => 'SQUAL_CHAR20',
```

```
  p_DISPLAY_TYPE => 'CHAR',
```

```
  p_SOURCE_COLUMN_NAME => 'ATTRIBUTE38',
```

```
  p_ATTR_RELATION_FACTOR => 227,
```

```
  p_CONVERT_TO_ID_FLAG => 'N',
```

```
  p_COMPARISON_OPERATOR_CN => 'q2026_cp',
```

```
  p_LOW_VALUE_CHAR_CN => 'q2026_low_value_char',
```

```
  p_HIGH_VALUE_CHAR_CN => NULL,
```

```
  p_LOW_VALUE_CHAR_ID_CN => NULL,
```

```
  p_LOW_VALUE_NUMBER_CN => NULL,
```

```

p_HIGH_VALUE_NUMBER_CN => NULL,
p_VALUE1_ID_CN => NULL,
p_VALUE2_ID_CN => NULL,
p_VALUE3_ID_CN => NULL,
p_VALUE4_ID_CN => NULL,
p_FIRST_CHAR_CN => NULL,
p_CURRENCY_CODE_CN => NULL,
p_EQUAL_FLAG => 'Y',
p_BATCH_OP_EQL => '( A.squal_char20 = B.q2026_low_value_char AND
B.q2026_cp = ''='' )',
p_LIKE_FLAG => 'N',
p_BATCH_OP_LIKE => NULL,
p_BETWEEN_FLAG => 'N',
p_BATCH_OP_BETWEEN => NULL,
p_BATCH_OP_COMMON_WHERE => NULL,
p_SEEDED_FLAG => NULL,
p_ALIAS_RULE1 => NULL,
p_REAL_TIME_SELECT => NULL,
p_REAL_TIME_WHERE => NULL,
p_REAL_TIME_FROM => NULL,
p_HTML_LOV_SQL1 => 'select /*+ FIRST_ROWS(10) */ 1 lookup_id, meaning
meaning, lookup_code lookup_code from cn_lookups where lookup_type =
''CN_EXAMPLE_LINE_OF_BUSINESS'' ',
p_HTML_LOV_SQL2 => NULL,
p_HTML_LOV_SQL3 => NULL,
p_HIERARCHY_TYPE => NULL,
retcode => l_retcode,
errbuf => l_errbuf);
END;
/

```

### 3. Update the BATCH\_TOTAL\_SQL to assign the new custom attribute.

---

#### Important

Run this SQL to see which fields are already enabled in the existing batch\_total\_sql's select statement.

```

sql> select batch_total_sql from cn_rs_usage_details_all where usage_id =
-1001 and org_id = &org_id ;

```

---

```

DECLARE
l_retcode VARCHAR2(30);
l_errbuf VARCHAR2(10000);
BEGIN
update cn_rs_usage_details_all
set batch_total_sql =
' SELECT T.TRANSACTION_ID TRANS_OBJECT_ID , ' ||
' T.PARTICIPANT_ID ASSIGNED_OBJECT_ID , ' ||
' T.SOURCE_EVENT_DATE EVENT_DATE , ' ||
' T.WORKER_ID , ' ||
' UPPER(substr(C.ACCOUNT_NAME,1,1)) SQUAL_FC01 , ' ||
' null SQUAL_CURC01 , ' ||
' T.SALES_CHANNEL SQUAL_CHAR01 , ' ||

```

```

' T.CITY SQUAL_CHAR02 , ' ||
' T.STATE SQUAL_CHAR03 , ' ||
' T.COUNTRY SQUAL_CHAR04 , ' ||
' C.ACCOUNT_NAME SQUAL_CHAR05 , ' ||
' P.PARTY_NUMBER SQUAL_CHAR06 , ' ||
' P.CATEGORY_CODE SQUAL_CHAR07 , ' ||
' T.POSTAL_CODE SQUAL_CHAR08 , ' ||
' T.PROVINCE SQUAL_CHAR09 , ' ||
' T.AREA_CODE SQUAL_CHAR10 , ' ||
' T.CUSTOMER_ID SQUAL_NUM01 , ' ||
' T.INVENTORY_ITEM_ID SQUAL_NUM02 , ' ||
' NULL SQUAL_NUM03 , ' ||
' T.SOURCE_ORG_ID SQUAL_NUM04 , ' ||
' T.PARTICIPANT_ID SQUAL_NUM05 , ' ||
' null SQUAL_FC02 , ' ||
' null SQUAL_CURC02 , ' ||
' NULL SQUAL_CHAR11 , ' ||
' T.TRANSACTION_TYPE SQUAL_CHAR12 , ' ||
' T.TERR_NAME SQUAL_CHAR13 , ' ||
' NULL SQUAL_CHAR14 , ' ||
' NULL SQUAL_CHAR15 , ' ||
' NULL SQUAL_CHAR16 , ' ||
' NULL SQUAL_CHAR17 , ' ||
' NULL SQUAL_CHAR18 , ' ||
' null SQUAL_CHAR19 , ' ||
' T.ATTRIBUTE38 SQUAL_CHAR20 , ' ||
' T.TRANSACTION_AMT_FUNC_CURR SQUAL_NUM06 , ' ||
' T.TRANSACTION_QTY SQUAL_NUM07 , ' ||
' T.ORG_ID SQUAL_NUM08 , ' ||
' null SQUAL_NUM09 , ' ||
' null SQUAL_NUM10 , ' ||
' null SQUAL_FC03 , ' ||
' null SQUAL_CURC03 , ' ||
' null SQUAL_CHAR21 , ' ||
' null SQUAL_CHAR22 , ' ||
' null SQUAL_CHAR23 , ' ||
' null SQUAL_CHAR24 , ' ||
' null SQUAL_CHAR25 , ' ||
' null SQUAL_CHAR26 , ' ||
' null SQUAL_CHAR27 , ' ||
' null SQUAL_CHAR28 , ' ||
' null SQUAL_CHAR29 , ' ||
' null SQUAL_CHAR30 , ' ||
' null SQUAL_NUM11 , ' ||
' null SQUAL_NUM12 , ' ||
' null SQUAL_NUM13 , ' ||
' null SQUAL_NUM14 , ' ||
' null SQUAL_NUM15 , ' ||
' null SQUAL_FC04 , ' ||
' null SQUAL_CURC04 , ' ||
' null SQUAL_CHAR31 , ' ||
' null SQUAL_CHAR32 , ' ||
' null SQUAL_CHAR33 , ' ||
' null SQUAL_CHAR34 , ' ||

```

```

' null SQUAL_CHAR35 , ' ||
' null SQUAL_CHAR36 , ' ||
' null SQUAL_CHAR37 , ' ||
' null SQUAL_CHAR38 , ' ||
' null SQUAL_CHAR39 , ' ||
' null SQUAL_CHAR40 , ' ||
' null SQUAL_NUM16 , ' ||
' null SQUAL_NUM17 , ' ||
' null SQUAL_NUM18 , ' ||
' null SQUAL_NUM19 , ' ||
' null SQUAL_NUM20 , ' ||
' null SQUAL_FC05 , ' ||
' null SQUAL_CURC05 , ' ||
' null SQUAL_CHAR41 , ' ||
' null SQUAL_CHAR42 , ' ||
' null SQUAL_CHAR43 , ' ||
' null SQUAL_CHAR44 , ' ||
' null SQUAL_CHAR45 , ' ||
' null SQUAL_CHAR46 , ' ||
' null SQUAL_CHAR47 , ' ||
' null SQUAL_CHAR48 , ' ||
' null SQUAL_CHAR49 , ' ||
' null SQUAL_CHAR50 , ' ||
' null SQUAL_NUM21 , ' ||
' null SQUAL_NUM22 , ' ||
' null SQUAL_NUM23 , ' ||
' null SQUAL_NUM24 , ' ||
' null SQUAL_NUM25 ' ||
' from CN_TP_TRANSACTIONS_ALL T , ' ||
' HZ_PARTIES P ' ||
' where T.CUSTOMER_ID = P.PARTY_ID(+) '
where usage_id = -1001;
END;
/

```

## Enabling Custom Attribute for Crediting

1. Go to the Define Business Unit Configuration for Incentive Compensation task list, Configure Tables and Columns page.
2. In the Tables section, select **CN\_TP\_TRANSACTIONS\_ALL**.
3. Scroll to your attribute and select **Enable for Crediting**.

If you want to enable the attribute for classification, then select **Enable for Classification**.

4. Click **Save and Close**.

---

### Tip

To expose your customer attribute in the user interface, and in application-generated spreadsheets, edit and deploy the descriptive flexfield.

---

## **FAQs for Manage Task Configurations**

### **How can I enable an attribute to show in the incentive compensation expression builder?**

In the Define Business Unit Configuration for Incentive Compensation task list, Configure Tables and Columns task, enable the attribute for calculation and select the appropriate level 2 expression grouping. Also add a user-friendly name for the attribute, that the application displays in the incentive compensation expression builder.

### **Why can't I enable an attribute for the incentive compensation Classification or Crediting process?**

You must first add the custom qualifiers to the incentive compensation classification or crediting rules.

---

## 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 your service administrator (Oracle Cloud implementations) or system administrator (other implementations) must download the library from Oracle Technology Network (<http://www.oracle.com/technetwork/documentation/default-1577155.html>) and put all the content from within the extracted folder (for example E28271\_01) directly into the appmgr/APPLTOP/fusionapps/applications/ahc/afh/reference/TechLib folder.

---

## 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. The collaboration features are also used elsewhere in Oracle Fusion Applications. Discussions may not be available in Oracle Cloud implementations.

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 users with the Oracle Fusion Help Text Administration duty role have access to the Manage Announcements icon button in the Announcements sections. They can create, edit, and delete announcements for the tab that they are on, and set the date range for when each announcement is to be displayed.

## **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.

---

Only users with the appropriate administrator role can moderate discussions. For more information on granting the administrator role on the discussions server, see the Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter Portal.

## **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.

Decisions to Consider	In This Example
What type of users do you need to limit help access to?	Human resources (HR) specialists
Is there a specific time period for which this access is needed?	No, the help files should always be viewed only by the HR specialists
Where do you want this group to appear in the list of values for help security groups?	First

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**.
2. On the Manage Help Security Groups page, add a new row.
3. Complete the fields, as shown in this table. Leave the start and end dates blank.

Field	Value
Help Security Group	HR
Meaning	HR Only
Description	Viewing by HR specialists only
Display Sequence	1

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.

---

If your enterprise has purchased Oracle User Productivity Kit (UPK) content, then your administrator can also add a **UPK** item to the **Help** menu in the global area of Oracle Fusion Applications. When users select this menu item, they access UPK content specific to the page that they are on.

## 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 title IDs that represent help files.

This figure provides an example of related links code.

```
OfaRelatedTopics(CREATE_AUTOMATIC_POSTING_CRITERIA_S_0000,  
JOURNAL_ENTRIES_HOW_THEY_RE_RECORDER_0000)
```

- You can delete this code to remove all related help, or delete title IDs to remove individual links (for example, `CREATE_AUTOMATIC_POSTING_CRITERIA_S_0000`).
- To replace existing links or add new links, you need to retain the code syntax and enter desired title IDs. To find title IDs, search for the help files on the Manage Custom Help page. Title IDs are displayed in the search results, but the **Title ID** column is hidden by default.

### 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:

```
<span class="HP_topic-link_bridgeDocument-linkToSTDoc_"><?ofa  
linkToSTDoc(WCSUG4636) ?><span class="HP_topic-linktext_">Understanding  
Tags</span><?ofa endLink ?></span>.
```

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.

```
OfaGlossaryTerm("accounting period", ACCOUNTING_PERIOD_0001)
```

In this example, `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 glossary term links, you must follow the link syntax and use the correct title ID for the glossary term. You can find title IDs in the search results of the Manage Custom 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.

Decisions to Consider	In This Example
What changes do you need to make to the guide?	Change the title of a chapter and remove a section in that chapter, to hide content about a particular subject
Which help window should the customized guide appear in?	The help window for the entire Welcome dashboard of Oracle Fusion Applications
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: <ul style="list-style-type: none"><li>• Search by Common Tasks navigator, under the Collaboration node</li><li>• Search by Business Process navigator, under <b>Information Technology Management - Manage Networking and Communications - Manage Social Networking Capabilities</b></li></ul>
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.

Field	Value
Title	The name of the UPK topic.
Source Type	Oracle User Productivity Kit
File Location	The full URL of the player package folder on the Web server, for example, <code>http://&lt;your domain&gt;.com/UPKcontent/PlayerPackage</code> .
Document ID	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.
Help Type	Demo
Help Security Group	Unsecured
Keywords	Terms relevant to the demo.
Description	Summary of the demo.
Include in New and Updated pane	Selected

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 Customize Help

### 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 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.

---

### Note

The Define Application Toolkit Configuration task list is available in implementation projects only if the Application Toolkit Component Maintenance feature choice is selected.

---

Use the Setup and Maintenance work area to access the tasks in the Define Application Toolkit Configuration task list.

## 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.

## 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.

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

Taxonomy Hierarchy page, which you can access by starting in the Setup and Maintenance Overview page and searching for the Manage Taxonomy Hierarchy task.

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: 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.

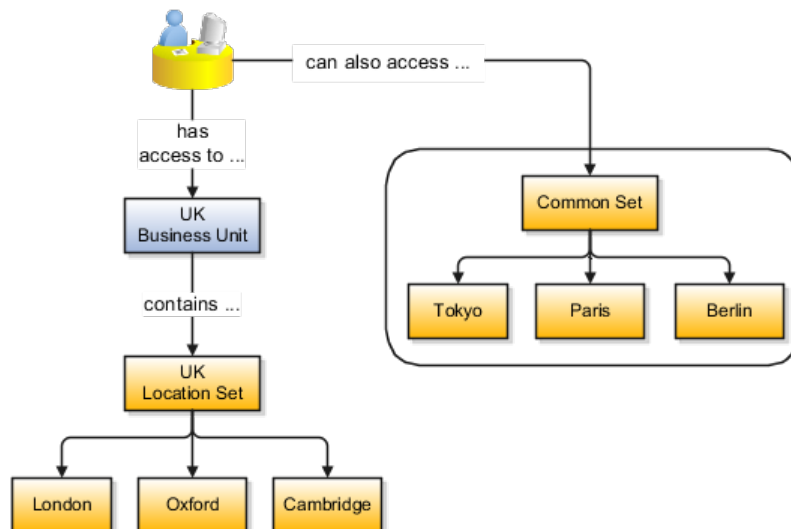
Consider the following scenario.

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 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.

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

## 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.

Lookup Type	Lookup Code	Meaning	Tag
MAR_STATUS	M	Married	
	S	Single	
	R	Registered Partner	+NL
	DP	Domestic Partner	-FR, AU

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.

Allowed Task	User	Extensible	System
Deleting a lookup type	Yes	No	No
Inserting new codes	Yes	Yes	No
Updating start date, end date, and enabled fields	Yes	Yes, only if the code is not predefined data	No
Deleting codes	Yes	Yes, only if the code is not predefined data	No
Updating tags	Yes	No	No
Updating module	Yes	No	No

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.

Lookup	Description
Standard	Lists the available codes and translated meanings
Set enabled	Additionally associates a reference data set with the lookup codes
Common	Legacy lookups

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.

Lookup type parameters	Value
Lookup type name	COLORS
Meaning	Status
Description	Status by color
Module	Oracle Fusion Middleware Extensions for Applications

The lookup codes you define for the COLORS lookup type are, BLUE, RED, GREEN, and YELLOW.

Lookup Code	Meaning	Enabled	Display Sequence
BLUE	Urgent	No	4
RED	Stop	Yes	1
GREEN	Go	Yes	3
YELLOW	Caution	Yes	2

### 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.

Displayed Value	Hidden ID
Stop	RED
Caution	YELLOW
Go	GREEN

### 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

Transaction number	User name	Status
1	Jane	RED
2	Bob	YELLOW
3	Alice	BLUE

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.

Reference Data Set	Lookup Code	Lookup Meaning
US	RED	Red
US	YELLOW	Yellow
US	GREEN	Green
EU	RED	Rouge
EU	ORANGE	Orange

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.

Reference Data Set	Lookup Code	Lookup Meaning
US	SEC	SEC Compliance Review
US	DIR	Needs Director's Approval
US	VP	Needs Vice President's Approval
CHINA	CSRC	Pending China Securities Regulatory Commission Review
CHINA	PR	Needs President's Approval
COMMON	REQUESTED	Customer Request

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.

---

Area of Difference	Lookup Type	Value Set
List of values	Static	Dynamic if Table validation type
Validation of values	One to one match of meaning to code included in a lookup view, or through the determinant of a reference data set	By format or inclusion in a table
Format type of values	char	varchar2, number, and so on
Length of value	Text string up to 30 characters	Any type of variable length from 1 to 4000
Duplication of values	Never. Values are unique.	Duplicate values allowed



Management	Managed by both administrators and end-users, except system lookups or predefined lookups at the system customization level, which cannot be modified.	Maintained by administrators, except some product flexfield codes, such as GL for Oracle Fusion General Ledger, which are maintained by end users
------------	--	---

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.

The message dictionary is described in the Oracle Fusion Applications Developer's Guide, and other messages in the Oracle Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework.

#### **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

See: Internationalizing and Localizing Pages

## 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

## 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.

### **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.

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

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.

Who	can do	what	on which set of data
Accounts payable managers	view	AP disbursements	for their business unit

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.

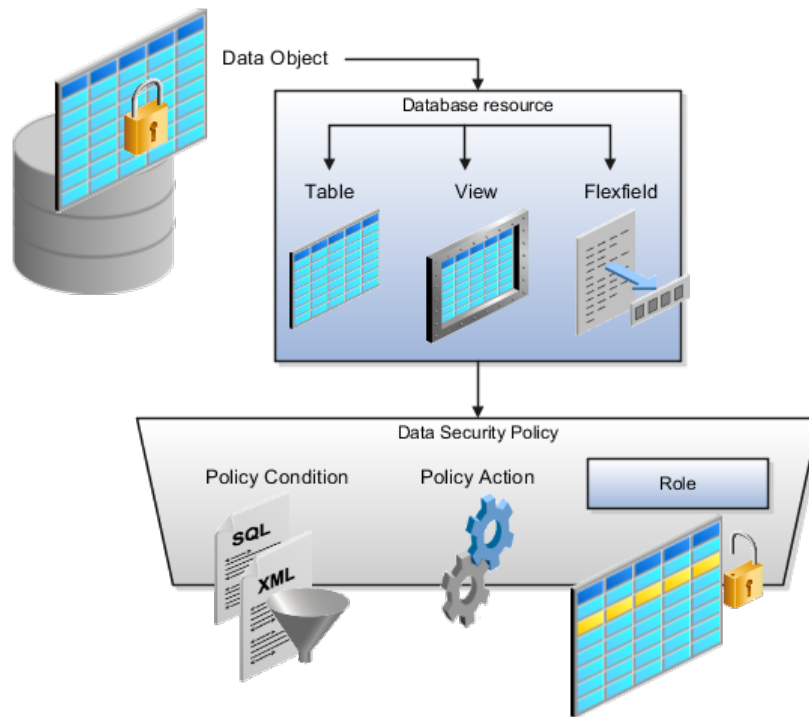
## **Database Resources and Data Security Policies: How They Work Together**

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.

Method of securing data	Reason	Example
Data roles apply explicit data security policies on job and abstract roles	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.	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.
Data security policies	Define data access for application roles and provide inheriting job and abstract roles with implicit data security	Projects

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, `REGION='WEST'` is a static object instance set and `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).

---

**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.

Status	Meaning
Draft	Yet to be published or is in a modified state.

Active	In use and based on which one or more trees or tree versions are created.
Inactive	Not in use.

## Tree Structure Audit Results: Explained

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

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

### Running an Audit

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

### Validation Details

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

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

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

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



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

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

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

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

### Labeling Schemes

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

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

### Restriction of Tree Node Values

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

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

### Data Source Values and Parameters

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

---

#### Note

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

---

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

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

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

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

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

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

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

### Row Flattening

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

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

### Column Flattening

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

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

### Column Flattened Entity Objects

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

### ADF Business Component View Objects

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

## Manage Tree Labels

### Tree Labels: Explained

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

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

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

## Manage Trees and Tree Versions

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

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

#### Creating and Editing Trees

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

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

#### Note

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

#### Creating and Editing Tree Versions

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

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

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

### Tree Version Audit Results: Explained

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

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

## Running an Audit

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

## Validation Details

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

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

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

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

Date Restricted Node	On the Manage Tree Structures: Specify Data Sources page, if the <b>Date Range</b> check box is selected to enable the <b>Restrict Tree Node List of Values Based on</b> option for a tree structure, each node in the underlying data source must have an effective date range same as the effective date range of the tree version. This restriction does not apply if the check box is not selected.	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).	Ensure that all nodes in the tree version have effective date range for the effective date range for the tree version.
Multiple Active Tree Version	On the Manage Tree Structures: Specify Definition page, if the <b>Allow Multiple Active Tree Versions</b> check box is not selected for the tree structure, there should not be more than one active tree version under a tree at any time. This restriction does not apply if the check box is selected.	Even when the check box is not selected, there is more than one active tree version in the tree for the same date range.	Set no more than one tree version to Active within the same date range and set the others to inactive or draft status.
Range Based Node	On the Data Source dialog box, if the <b>Allow Range Children</b> check box is not selected, range-based nodes are not permitted from that data source. This restriction does not apply if the check box is selected.	Even when the check box is not selected, there are range-based nodes from a data source.	Ensure that any range nodes in your tree version are from a data source that allows range children.



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

## Trees and Data Sources: How They Work Together

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

### Metadata

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

### Data Storage

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

### Access Control

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

## Adding Tree Nodes: Points to Consider

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

usually have a storage home, only user-created data needs to be stored in an entity.

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

## Managing Tree Nodes

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

## Node Levels

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

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

## Node Types

A tree node has the following node types.

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

# 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.

Type of Information	Profile Option Example
User preferences	Settings to provide access to social networking features
Installation information	Setting to identify the location of a portal
Configuration choices	Settings to change user interface skins and behaviors
Processing options	Settings to affect how much information to log either for an entire site or a specific user

You can add and configure new profile options in addition to configuring predefined profile options that are implemented as updateable.

### 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.

Profile Option Level	Value of the Profile Option Level	Profile Option Value	Effect
User	Manager1	UK pound sterling	Access to site and all products shows UK pounds sterling in effect

User	Manager2	US dollar	Access to site and all products shows US dollars in effect
Product	Financials for EMEA	Euro	Unless superseded by a user level value, Euros in effect for Financials for EMEA applications
Site	Site	UK pound sterling	UK pounds sterling in effect for all other users and products

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.

Hierarchy Level	Priority When Multiple Levels Set	Effect on Applications	Example
Site	Lowest	Affect all applications for a given implementation	Currency for the site is set to Euros.
Product	Supersedes Site	Affect all applications of a product family such as Financials	Currency for the Financials products set to UK pound sterling.
User	Highest, supersedes Product	Affect only the experience of the current user	Currency for the user of Financials applications set to US dollars.

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.

---

## Incentive Compensation Profile Options: Explained

Oracle Fusion Incentive Compensation has two profile options, which you can set at the site level.

- **CN\_DEBUG:** Select **Yes** when you want the application to write the debug errors to the CN\_PROCESS\_AUDIT\_LINES table.
- **CN\_MARK\_EVENTS:** Select **Yes** when you want the application to record every event in the Changed Events Log for inclusion in the next incremental calculation and plan deployer processes.

## 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.

Tasks	Role	Considerations
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.

Do not use profile options to capture a dynamic system states, such as data stored in a temporary table. Use Global Variables for temporary states instead.

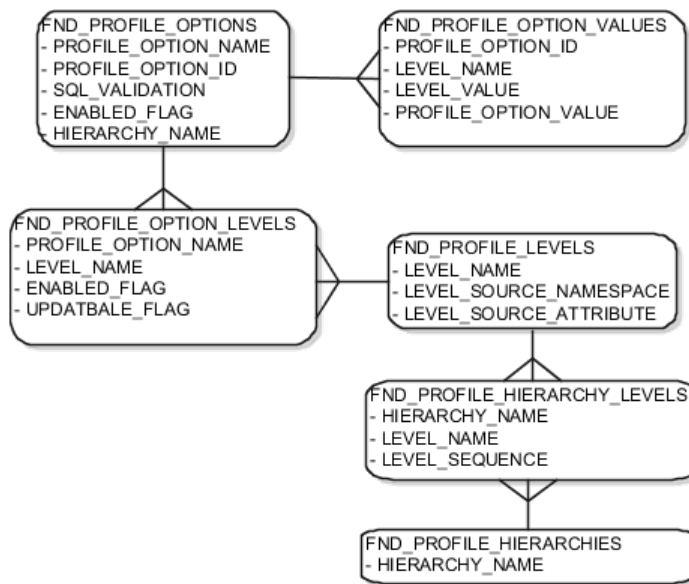
Evaluate if there is a genuine need before creating a profile option. Do not force users to make a decision about an aspect of their application use that is of no concern.

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\_LANUGUAGE\_VL.

Display Value	Hidden Value
American English	US
French	F
Spanish	E

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.

Level Name	Level Value	Profile Option Value
Site	InFusion	American English
Product	Customer Center	French
Product	CRM Application Composer	American English
User	Application Administrator	American English
User	Hima	Hindi

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.

Site	Product	User	Highest Available Level	Active Profile Option Value
InFusion	CRM Application Composer	Hima	User	Hindi
Acme	Payables	Application Administrator	User	American English
InFusion	Customer Center	Guillaume	Product	French
InFusion	Payables	Implementation Consultant	Site	American English
Acme	Payables	Implementation Consultant	none	no value

---

#### 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

Flexfields are a mechanism for configuring and extending applications to meet enterprise-specific needs. Using flexfields, you can extend application objects to capture data that would not otherwise be tracked by the application, or configure intelligent key codes comprised of meaningful parts according to your business practices. 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.

A flexfield is an extensible data field consisting of subfields. Each subfield is a flexfield segment. You configure segments to capture enterprise-specific information. Each segment captures a single atomic value, has a name, and maps to a pre-reserved column in the applications database.

Flexfields allow enterprise requirements to be met without changing the data model. Different data can be captured on the same database table.

Application developers create a flexfield and register it so that it is available for configuration. Administrators and implementation consultants setup or configure segments and other properties of the available flexfields. End users see flexfield segments as attributes of information displayed in the application user interface. They enter a value for the attribute. The value may be selected from a list of valid values or entered as free-form text that complies with formatting rules.

Manage flexfields using tasks you access by starting in the Setup and Maintenance Overview page.

---

#### Tip

Use the **Business Object** parameter to search Application Key Flexfields, Application Descriptive Flexfields, and Application Extensible Flexfields to find all tasks related to flexfields. To manage any flexfield across all Oracle Fusion Applications products, search for the Define Flexfields task list and access the

Manage Descriptive Flexfields, Manage Extensible Flexfields, and Manage Key Flexfields tasks.

---

## Types of Flexfields

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

For example, in Oracle Fusion Financials, key flexfields represent objects such as accounting codes and asset categories. Generally, correct operations of a product depend on key flexfield setup. In Oracle Fusion Payables, a descriptive flexfield expands a customizable area of an invoices page by providing subfields for collecting invoice details. You can implement these subfields, which are descriptive flexfield segments, as context sensitive so they appear only when needed. Extensible flexfields are similar to descriptive flexfields, but provide additional advanced features. Generally setup of descriptive and extensible flexfields is optional because their segments capture additional information that is stored attributes on an entity

## 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.

You define the appearance and meaning of individual segments when configuring a flexfield.

A key flexfield segment commonly describes a characteristic of the entity identified by the flexfield, such as a part number structured to include information about the type, color, and size of an item. A descriptive flexfield segment represents an attribute of information that describes a characteristic of the entity identified on the application page, such as details about a device containing components, some of which are globally present on the page while others are contextually dependent on the category of the device.

## Value Sets

A value set is a predefined, named group of values that can be used to validate the content of a flexfield segment.

You configure a flexfield segment with a value set that establishes the valid values that an end user can enter for the segment. You define the values in a value set, including such characteristics as the length and format of the values. You can specify formatting rules, or specify values from an application table or predefined list. Multiple segments within a flexfield, or multiple flexfields, can share a single value set.

## Structure and Context

Key flexfields have structure. Descriptive flexfields and extensible flexfields have context.

Each key flexfield structure is a specific configuration of segments. Adding or removing segments, or rearranging their order, produces a different structure. The database columns on which segments in different structures are based can be reused in as many structures as desired.

Descriptive flexfield segments can be context-sensitive, which means available to an application based on a context rather than globally available wherever the flexfield appears. (which means available to an application based on a context value rather than globally available wherever the flexfield appears. A descriptive flexfield context is a set of context-sensitive segments that store information related to the same context value. You define contexts as part of configuring a descriptive flexfield. End users see global segments, as well as any context-sensitive segments that apply to the selected context value.

Extensible flexfield segments are made available to an application based upon a category value. An extensible flexfield context serves as a container for related segments, used to organize the various segments that are applicable to a category value. You define contexts with context-sensitive segments and associate them to categories as part of configuring an extensible flexfield. End users see the segments displayed in subregions, one for each context associated to the selected category value.

In descriptive flexfields and extensible flexfields, the database columns on which context-sensitive segments are based can be reused in as many contexts as desired.

## 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 appear in the UI based on the flexfield definition at the time of last deployment, rather than the current definition.

You can deploy a flexfield as a sandbox for testing the configuration before deploying it to the mainline for all users.

## Runtime Appearance

In an application user interface, descriptive 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.

Use the **Highlight Flexfields** command in the Administration menu of the Setup and Maintenance work area to identify the location of the descriptive and key flexfields on the runtime page. Flexfields in highlight mode display an information icon to access details about the flexfield, and an edit icon to manage the flexfield.

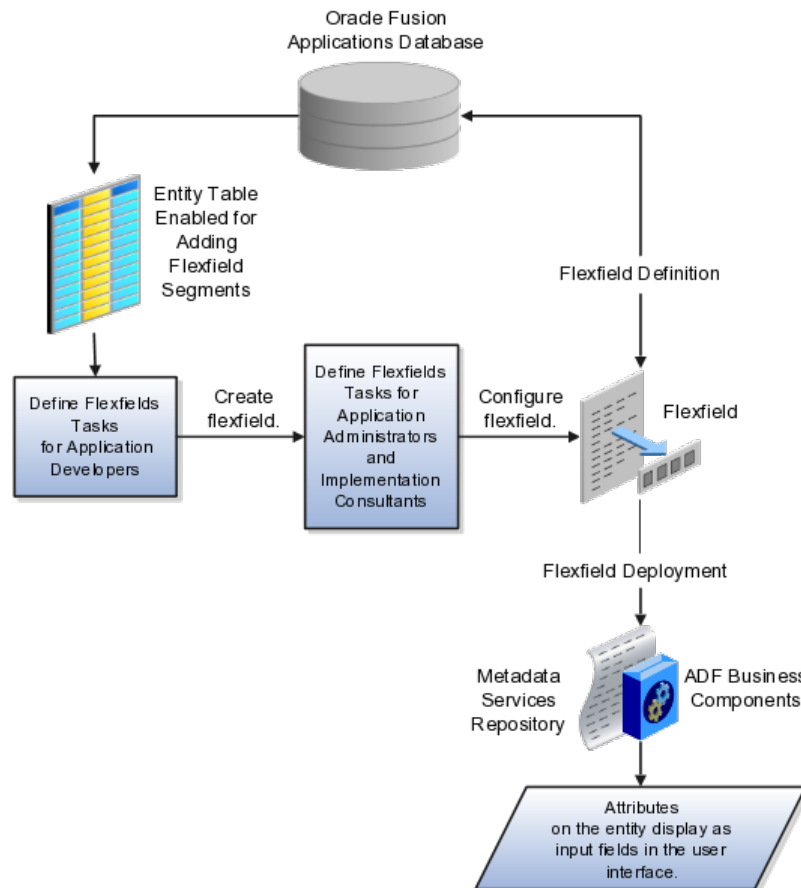
All segments of a single flexfield are grouped together by default. The layout and positions of the flexfield segments depend on where the application developer places the flexfield on the page. Flexfields may also be presented in a separate section of the page, in a table, or on their own page or subwindow.

You can use Oracle Composer to edit the layout, position, or other display features of the flexfield segments.

## 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.

## Integration

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 business intelligence (Oracle Business Intelligence, Extended Spread Sheet Database (ESSbase)).

Flexfield configurations 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.

## Importing and Exporting

You can export and import flexfields with a deployment status of Deployed or Deployed to Sandbox across instances of Oracle Fusion Applications using the Setup and Maintenance Overview page. Ensure a flexfield is eligible for migration (by verifying that it has successfully deployed) prior to attempting the migration.

## 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.

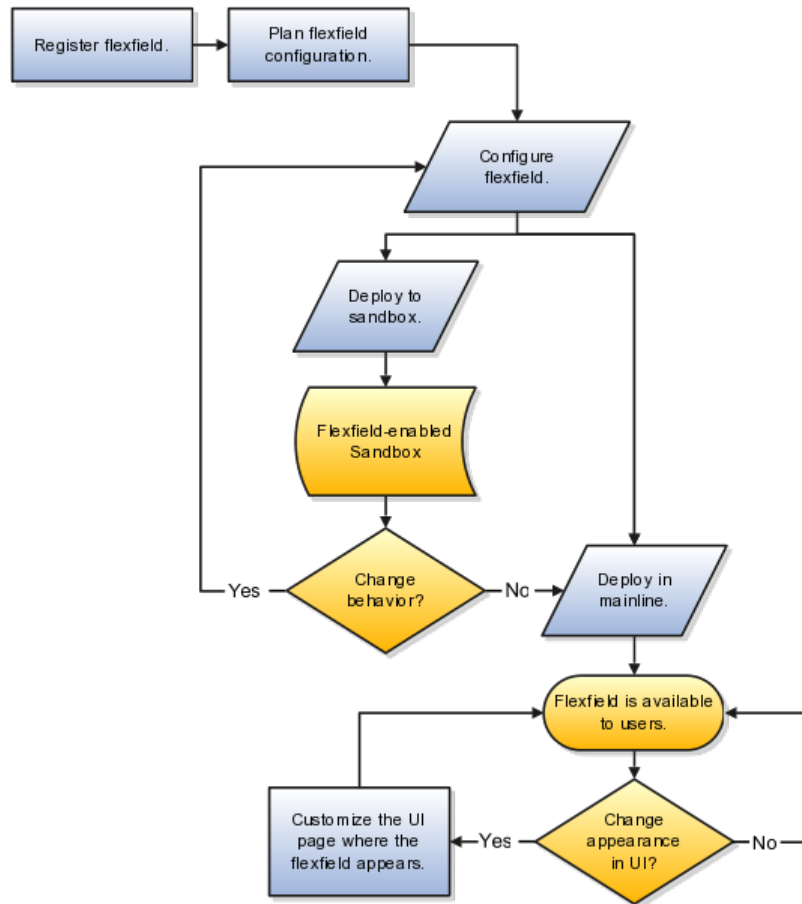
## Flexfield Management

### 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 flexfields provide parameters that 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 Segment for Business Intelligence

You can enable key flexfield 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 when enabling key and descriptive flexfield segments for business intelligence.

## Deploying Flexfields

After you configure or change 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.

### **Identifying Flexfields on a Runtime Page**

The **Highlight Flexfields** command in the Administration menu of the Setup and Maintenance work area identifies the location of descriptive and key flexfields on the runtime page by displaying an information icon for accessing details about each flexfield.

Even if a flexfield has not yet been deployed and does not appear on the runtime page in normal view, it appears in the Highlight Flexfield view for that page. **Highlight Flexfields** accesses the current flexfield metadata definition.

Use the highlighted flexfield's edit icon to manage the descriptive or key flexfield directly. Alternatively, note a highlighted flexfield's name to search for it in the tasks for managing flexfields.

## **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.

## Code Naming Conventions

When entering codes such as segment code, enter a code consisting of the characters A-Z, a-z, 0-9, with a non-numeric leading character.

The code is used in the flexfield's element in the XML schema for web services. You can maximize the readability of the schema by naming codes with a leading alphabetic character followed by alphanumeric characters. The use of spaces, underscores, multi-byte characters, and leading numeric characters, which are all encoded in XML schemas, make the codes in the schema element difficult to read.

This applies to context code in descriptive and extensible flexfields, and to structure instance code in key flexfields.

## 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.

Default Type	Default value specified?	Derivation value specified?	Initial runtime behavior	Runtime behavior after parameter changes
None	No	Yes	No initial segment value	Changed parameter derivation value updates segment value
Constant	Yes	No	Default segment value	N/A
Constant	Yes	Yes	Default segment value	Changed parameter derivation value updates segment value
Parameter	Yes	No	Default segment value is the parameter's default value	N/A

Parameter	Yes	Yes, and same as default value	Default segment value is parameter's default and derivation value	Changed parameter derivation value updates segment value
Parameter	Yes	Yes, and different from default value	Default segment value is parameter's default value	Changed parameter default value does not update segment value. Only the changed derivation value updates the segment value.

## Flexfield Usages: Explained

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.

### Extensible Flexfields

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.

### Value Sets

The usage of value sets specifies the segments where the value set is assigned.

## 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.

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

### Deployment Status

Every flexfield has a deployment status.

A flexfield can have the following deployment statuses.

Deployment Status	Meaning
Edited	The flexfield metadata definition has not been deployed yet. Updates of the metadata definition are not applied in the runtime environment yet.
Patched	The flexfield metadata definition has been modified through a patch or through a data migration action, but the flexfield has not yet been deployed so the updated definition is not reflected in the runtime environment.
Deployed to Sandbox	The current metadata for the flexfield is deployed in ADF artifacts and available as a flexfield-enabled sandbox. The status of the sandbox is managed by the Manage Sandboxes task available to the Administrator menu of the Setup and Maintenance work area.
Deployed	The current metadata for the flexfield is deployed in ADF artifacts and available to end users. There have not been any changes to the flexfield since it was last deployed in the mainline.
Error	The deployment attempt in the mainline failed.

---

#### Note

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

---

### Initial Deployment Status of Flexfields

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

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

### Metadata Validation

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

## Flexfield Deployment Status: How It Is Calculated

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

### Settings That Affect Flexfield Deployment Status

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

### How Flexfield Deployment Status Is Calculated

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

---

#### Note

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

---

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

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

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

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

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

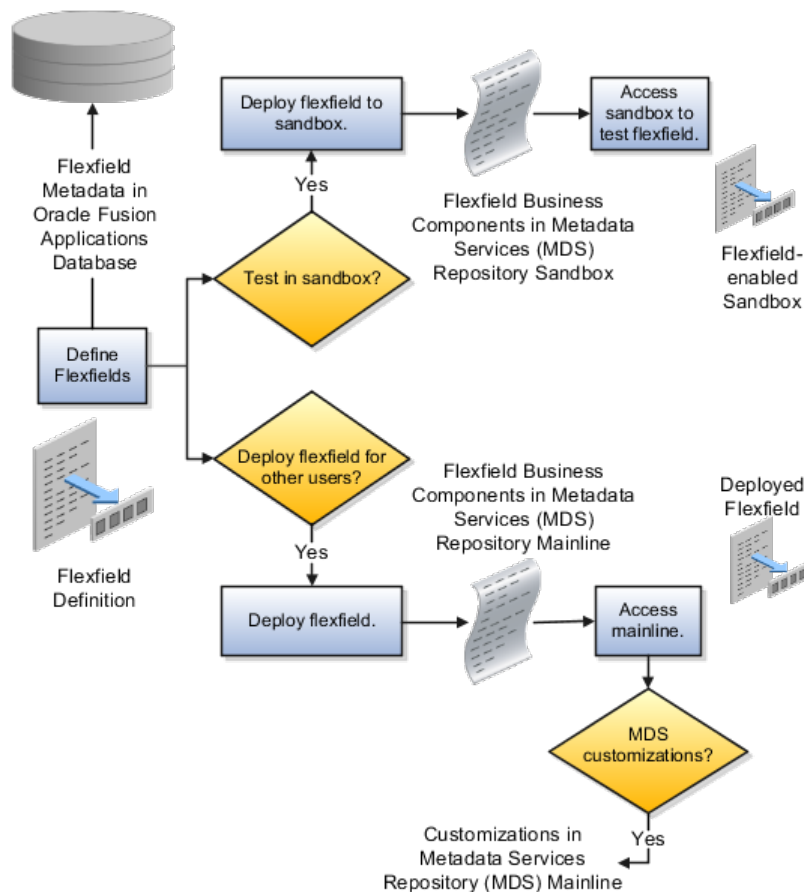
## Deploying a Flexfield-Enabled Sandbox: How It Works With Mainline Metadata

The flexfield definition in a sandbox corresponds to the flexfield metadata definition in the Oracle Fusion Applications database at the time the flexfield was deployed to the sandbox. When the flexfield is ready for end users, the flexfield must be deployed to the mainline.

A flexfield-enabled sandbox uses the following components.

- Flexfield metadata in the Oracle Fusion Applications database
- Flexfield business components in a sandbox Metadata Services (MDS) repository
- User interface customizations for the flexfield in the mainline MDS repository

The figure shows the two types of deployment available in the Manage Flexfields activity. Deploying a flexfield to a sandbox creates a sandbox MDS repository for the sole purpose of testing flexfield behavior. The sandbox is only accessible to the administrator who activates and accesses it, not to users generally. Deploying a flexfield to the mainline applies the flexfield definition to the mainline MDS repository where it is available to end users. After deploying the flexfield to the mainline, customize the page where the flexfield segments appear. Customization of the page in the sandbox MDS repository cannot be published to the mainline MDS repository.





## **Sandbox Metadata Services Repository Data**

Deploying the flexfield to a sandbox generates the Application Development Framework (ADF) business components of a flexfield in a sandbox MDS repository for testing in isolation.

---

### **Warning**

Do not make changes to flexfield segment display features in a flexfield-enabled sandbox as these changes will be lost when deploying the flexfield to the mainline.

---

## **Mainline Metadata Services Repository Data**

The Oracle Fusion Applications database stores the single source of truth about a flexfield. From this the ADF business component objects that implement the flexfield in the runtime user interface are generated in the mainline MDS repository when the flexfield is deployed.

## **Deploying a Flexfield-Enabled Sandbox: Points to Consider**

Deploying a flexfield to a sandbox creates a flexfield-enabled sandbox . Each flexfield-enabled sandbox contains only one flexfield.

You can test the runtime behavior of a flexfield in the flexfield-enabled sandbox. If changes are needed, return to the Define Flexfield tasks to change the flexfield definition.

When you deploy a flexfield to sandbox, the process reads the metadata about the segments from the database, generates flexfield Application Development Framework (ADF) business component artifacts based on that definition, and stores in the sandbox only the generated artifacts derived from the definition.

## **Sandbox MDS Repository Data**

The sandbox data allows you to test the flexfield in isolation without first deploying it in the mainline where it could be accessed by users.

---

### **Warning**

Do not make changes to flexfield segment display features in a flexfield-enabled sandbox as these changes will be lost when deploying the flexfield to the mainline.

---

## **Managing a Flexfield-Enabled Sandbox**

When you deploy a flexfield as a sandbox, that flexfield-enabled sandbox automatically gets activated in your user session. When you sign back in to see the changes, the sandbox is active in your session.

You can only deploy a flexfield to a sandbox using the Define Flexfields task flow pages.

You also can use the Manage Sandboxes feature in the Administration menu of the Setup and Maintenance work area to activate, access, or delete a flexfield-enabled sandbox.

---

**Note**

Whether you use the Define Flexfields or Manage Sandboxes task flows to access a flexfield-enabled sandbox, you must sign out and sign back in before you can see the changes you deployed in the runtime.

---

You cannot publish the flexfield from the sandbox to the mainline. You must use the Define Flexfields task flow pages to deploy the flexfield for access by users of the mainline because the flexfield configuration in the mainline is the single source of truth.

## Deploying Flexfields Using the Command Line: Explained

You can use the Manage Key Flexfields, Manage Descriptive Flexfields, and Manage Extensible Flexfields tasks to deploy flexfields. You can also use WebLogic Server Tool (WLST) commands for priming the Metadata Services (MDS) repository with predefined flexfield artifacts and for deploying flexfields.

The table describes the available commands.

WebLogic Server Tool Command	Description
<code>deployFlexForApp</code>	Deploys all flexfields for the specified enterprise application. Only flexfields whose status is other than deployed are affected by this command unless the option is enabled to force all flexfields to be deployed regardless of deployment status.  Initial application provisioning runs this command to prime the MDS repository with flexfield artifacts.
<code>deployFlex</code>	Deploy a single flexfield regardless of deployment status
<code>deployPatchedFlex</code>	Deploys flexfield changes that have been delivered using a flexfield Seed Data Framework (SDF) patch. Deploys flexfields that have a Patched deployment status.
<code>deleteFlexPatchingLabels</code>	Displays MDS label of flexfield changes for viewing and deleting patching labels.

Executing these commands outputs a report at the command line. The report provides the following information for every flexfield that is processed.

- Application identity (APPID)
- Flexfield code
- Deployment result, such as success or error

In case of errors, the report lists the usages for which the errors were encountered. If a runtime exception occurs, the output displays the traceback

information. For each WLST flexfield command, adding the `reportFormat='xml'` argument returns the report as an XML string.

Consider the following aspects of command line deployment.

- Preparing to use the WLST flexfield commands
- Using the `deployFlexForApp` command
- Using the `deployFlex` command
- Using the `deployPatchedFlex` command
- Using the `deleteFlexPatchingLabels` command
- Exiting the WLST and checking the results

## Preparing To Use the WLST Flexfield Commands

You can only execute the WLST flexfield commands on a WebLogic Administration Server for a domain that has a running instance of the Oracle Fusion Middleware Extensions for Applications (Applications Core) Setup application.

For more information on deploying the Applications Core Setup application, see the Oracle Fusion Applications Developer's Guide.

Ensure that the AppMasterDB data source is registered as a JDBC data source with the WebLogic Administration Server and points to the same database as the ApplicationDB data source.

Start the WebLogic Server Tool (WLST) tool, if it is not currently running.

UNIX:

```
sh $JDEV_HOME/oracle_common/common/bin/wlst.sh
```

Windows:

```
wlst.cmd
```

Connect to the server, replacing the user name and password arguments with your WebLogic Server user name and password.

```
connect('wls_username', 'wls_password', 'wls_uri')
```

The values must be wrapped in single-quotes. The `wls_uri` value is typically `T3://localhost:7101`.

For more information on the WLST scripting tool, see the Oracle Fusion Middleware Oracle WebLogic Scripting Tool.

## Using the `deployFlexForApp` Command

The `deployFlexForApp` command translates the product application's predefined flexfield metadata into artifacts in the MDS repository.

---

### Important

This command is run automatically when you provision applications. However, after custom applications development, you must run the `deployFlexForApp` command after you configure your application to read the flexfield artifacts from the MDS repository and before you log into the application for the first time, even if there is no predefined flexfield metadata.

---

This command does not deploy flexfields that have a status of Deployed unless the force parameter is set to `'true'` (the default setting is `'false'`).

For more information on priming the MDS partition with configured flexfield artifacts, see the Oracle Fusion Applications Developer's Guide.

From the WLST tool, execute the following commands to deploy the artifacts to the MDS partition, replacing `product_application_shortcode` with the application's short name wrapped in single-quotes.

```
deployFlexForApp('product_application_shortcode'[, 'enterprise_id']  
[, 'force'])
```

In a multi-tenant environment, replace `enterprise_id` with the Enterprise ID to which the flexfield is mapped. Otherwise, replace with `'None'` or do not provide a second argument.

To deploy all flexfields regardless of their deployment status, set force to `'true'` (the default setting is `'false'`). If you want to deploy all flexfields in a single-tenant environment, you either can set `enterprise_id` to `'None'`, or you can use the following signature:

```
deployFlexForApp(applicationShortName='product_application_shortcode', force='true')
```

---

### Tip

The application's short name is the same as the application's module name.

---

For more information about working with application taxonomy, see the Oracle Fusion Applications Developer's Guide.

## Using the `deployFlex` Command

From the WLST tool, execute the following command to deploy a flexfield, replacing `flex_code` with the code that identifies the flexfield, and replacing `flex_type` with the flexfield's type, which is either DFF, KFF, or EFF.

```
deployFlex('flex_code', 'flex_type')
```

The values must be wrapped in single-quotes.

## Using the `deployPatchedFlex` Command

Use the `deployPatchedFlex` command for situations where the patching framework does not invoke the command, such as when an application has been patched offline.

If the installation is multi-tenant enabled, the command deploys all patched flexfields for all enterprises. This command is not intended to be invoked manually.

Check with your provisioning or patching team, or the task flows for managing flexfields, to verify that the flexfield has a Patched deployment status.

From the WLST tool, execute the following command to deploy the artifacts to the MDS partition of all flexfields that have a READY status.

```
deployPatchedFlex()
```

Execute the following command to deploy all flexfields that have either a READY status or an ERROR status.

```
deployPatchedFlex(mode='RETRY')
```

## Using the deleteFlexPatchingLabels Command

Whenever you deploy flexfield changes to MDS using the `deployPatchedFlex()` WLST command, an MDS label is created in the format `FlexPatchingWatermarkdate+time`. Use the `deleteFlexPatchingLabels` command to inquire about and delete these labels.

From the WLST tool, execute the `deployPatchedFlex()` command with no arguments to delete the flexfield patching labels.

To output a list of flexfield patching labels, execute the command with the `infoOnly` argument, as follows:

```
deleteFlexPatchingLabels(infoOnly='true')
```

## Exiting the WLST and Checking the Results

To exit the tool, execute the following command.

```
disconnect()
```

Optionally, sign into the application, access user interface pages that contain flexfields, and confirm the presence of flexfields for which configuration exists, such as value sets, segments, context, or structures.

## Manage 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.

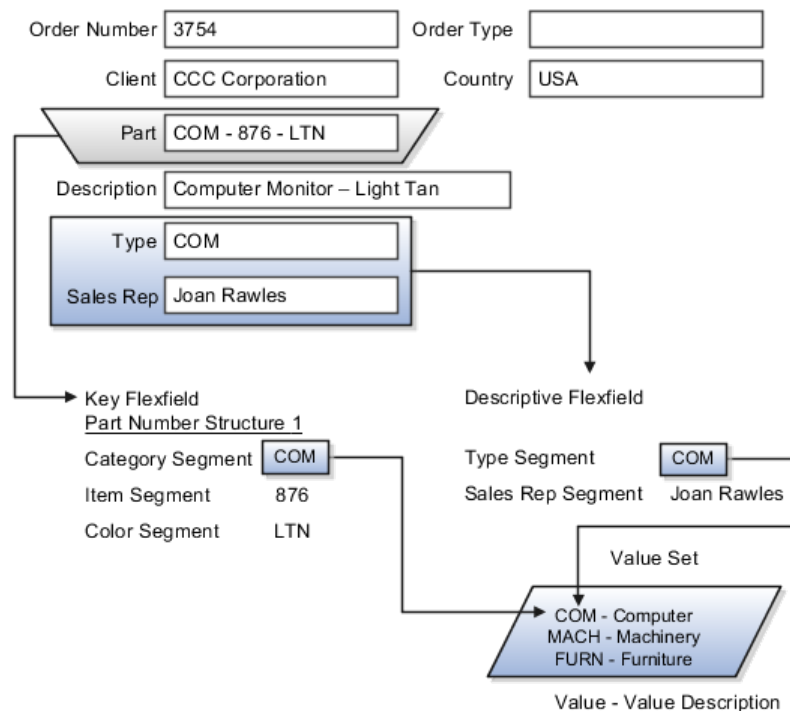
## Precision and Scale

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 optional. You can choose to configure and expose segments for 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.

#### Context

A descriptive flexfield can have only one context segment to provide context sensitivity.

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.

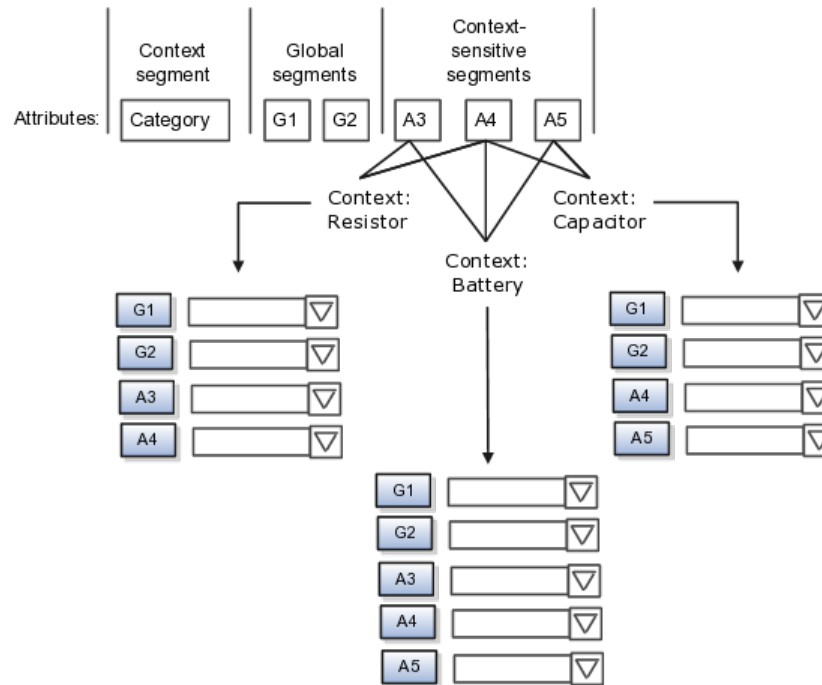
#### Segments and Contexts

Descriptive flexfield segments are of the following types.

Segment Type	Runtime Behavior
Global segment	Always available
Context segment	Determines which context-sensitive segments are displayed
Context-sensitive segment	Displayed depending on the value of the context segment

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.

## Incentive Compensation Flexfields: Explained

All standard, enabled incentive compensation descriptive flexfields relate to transactions. Use them to map transaction attributes in the source application, which do not exist by default in Oracle Fusion Incentive Compensation, to the CN\_TP\_TRANSACTIONS\_ALL table.

The descriptive flexfields consist of two data types: VARCHAR2 and NUMBER

- VARCHAR2 (240 CHAR): ATTRIBUTE1 -- ATTRIBUTE150
- NUMBER: ATTRIBUTE\_NUMBER1 -- ATTRIBUTE\_NUMBER50

Enable these attributes for use in the Crediting, Classification, and Calculation processes in the Define Business Unit Configuration for Incentive Compensation task list, Configure Tables and Columns task.

## 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

You can assign sequence order numbers to global segments and to context-sensitive segments in each context. 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.

## Enabling Descriptive Flexfield Segments for Business Intelligence: Points to Consider

A descriptive flexfield that is registered in the database as enabled for Oracle Business Intelligence (BI) includes a BI Enabled setting for each of its segments. When a global, context, or context-sensitive segment is BI-enabled, it is available for use in Oracle Business Intelligence.

The following aspects are important in understanding BI-enabled flexfield segments.

- Flattening business components to use BI-enabled segments in Oracle BI.
- Equalizing context-sensitive segments to prevent duplication and complexity in the flattened components
- Mapping attributes of flattened business components to logical objects in Oracle BI.
- Managing the labels that map segments to logical objects in Oracle BI.

After you deploy a business intelligence-enabled flexfield, import the flexfield changes into the Oracle Business Intelligence repository to make use of the newly-generated attributes in business intelligence applications. For additional information about import, refer to the Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition).

### Flattening

When you deploy a business intelligence-enabled descriptive flexfield, the deployment process generates an additional set of flattened business components for use in Oracle BI. The flattened business components include attributes for business intelligence-enabled segments only.

Flattened components include one attribute for the BI-enabled context-segment, and one attribute for each business intelligence-enabled global segment. For BI-enabled context-sensitive segments, if you assigned a label to the segment, the flattened components include an additional single attribute representing segments with that label. If you did not assign a label, the flattened components include a discrete attribute for each BI-enabled context-sensitive segment in each context.

### Equalization

You can prevent duplication and the extra workload and complexity that result from the flattening process by assigning a label to any set of context-sensitive segments that serve the same purpose in different contexts. BI-enabled context-sensitive segments that have labels are equalized so that the flattened business components include one attribute for each label, consolidated across contexts. Non-labeled context-sensitive segments are not equalized across context values, so the flattened components include a separate attribute for each context-sensitive segment for each context value.

---

#### Note

It may not be possible to equalize similarly labeled segments if they have incompatible data types or value set types.

---

## **Mapping to Logical Objects in Business Intelligence**

Assign a label to a global segment, context segment, or context-sensitive segment to map the corresponding attribute in the flattened components to a logical object in Oracle Business Intelligence. Using labels to map segments to BI logical objects minimizes the steps for importing the flexfield into Oracle Business Intelligence.

---

### **Note**

Assigning a label to a context-sensitive segment serves to equalize the attribute across contexts, as well as map the equalized attribute to business intelligence.

---

## **Managing Labels**

You may assign a predefined label (if available) to segments or create new labels for assignment, as needed. Specify a code, name, and description to identify each label. In the BI Object Name field, enter the name of the logical object in Oracle Business Intelligence to which the segment label should map during import. Specifying the BI logical object minimizes the steps for importing the flexfield into Oracle Business Intelligence and helps to equalize context-sensitive segments across contexts.

If no labels are assigned to a BI-enabled segment, or the BI Object Name on the assigned label does not exist in business intelligence, you must manually map the segment to the desired logical object when importing into Oracle Business Intelligence.

In addition, context-sensitive segments without labels cannot be equalized across context values. The flattened components include a separate attribute for each non-labeled context-sensitive segment in each context.

## **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**

Extensible flexfields support multiple contexts, and the contexts can be grouped into categories. Every extensible flexfield has at least one category, or root category, and some extensible flexfields support a hierarchy of categories, where a given category can inherit contexts from its parent 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.

In some applications, different users need to see different segment structures for the same flexfield. A key flexfield can have multiple structures if registered to support more than one structure.

The flexfield can display different fields for different end users based on a data condition in your application data, such as the value of another field entered by the end user or the user's role. For example, the correctly formatted local postal address for customer service inquiries differs based on locale. A postal address key flexfield could display different segments and prompts for different end users based on a location condition in your application data, such as the user's role or a value entered by the user.

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.

Each structure may have one or more structure instances. 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.

## **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.

Level Of Dynamic Combination Creation	Controlled By:
Flexfield	Application development
Each usage or reference to the key flexfield	Application development
Structure instance	Administrators and implementation consultants
Other	Administrators and implementation consultants

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.

---

### Structure Delimiters

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.

### 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.

---

Enable segments to indicate that they are in use. A flexfield does not display disabled segments in runtime.

---

**Tip**

To protect the integrity of your data, disable a segment if you have already used it to enter data.

---

## 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

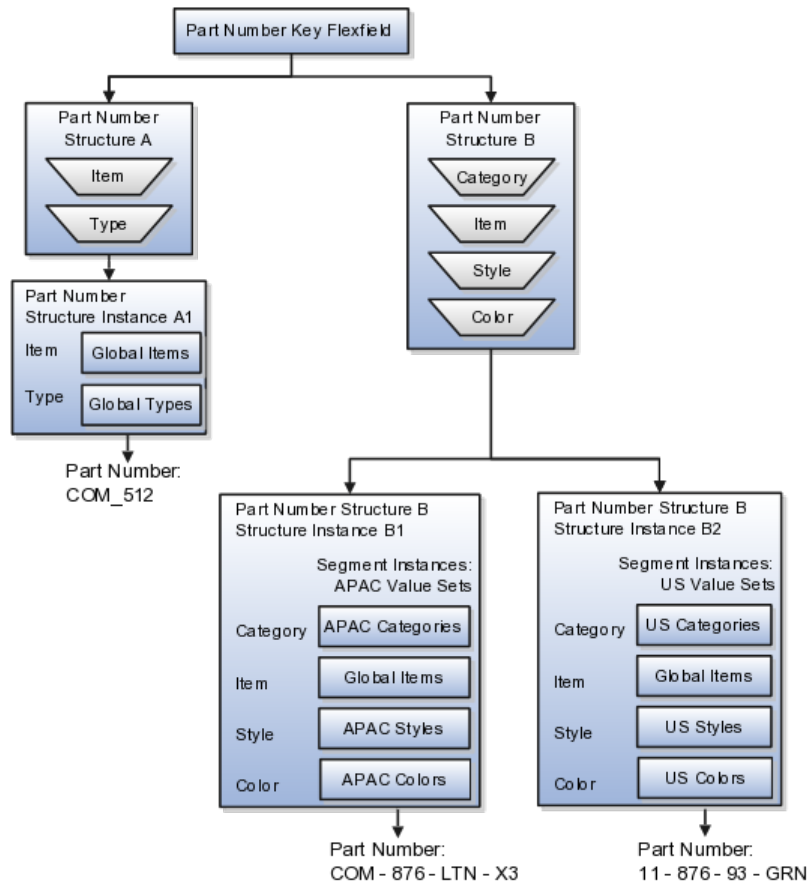
Differences among structure instances at the structure level include whether dynamic combination creation is allowed.

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.

## Enabling Key Flexfield Segments for Business Intelligence: Points to Consider

A key flexfield that is registered in the database as enabled for Oracle Business Intelligence (BI) includes a BI Enabled setting for each of its segment instances. When a segment instance is BI-enabled, it is available for use in Oracle Business Intelligence.

The following aspects are important in understanding BI-enabled key flexfield segments.

- Flattening business components to use BI-enabled segments in Oracle BI.
- Equalizing segments to prevent duplication and complexity in the flattened component.
- Mapping attributes of flattened business components to logical objects in Oracle BI.
- Managing the labels that map segments to logical objects in Oracle BI.

After you deploy a business intelligence-enabled flexfield, import the flexfield changes into the Oracle Business Intelligence repository to make use of the newly generated attributes in business intelligence applications. For additional information about import, refer to the Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition).

## Flattening

When you deploy a business intelligence-enabled key flexfield, the deployment process generates an additional set of flattened business components for use in business intelligence. The flattened business components include attributes for business intelligence-enabled segment instances only.

If you assigned a label to a segment, the flattened components include a single attribute representing all segment instances with that label. If you did not assign a label, the flattened components include a discrete attribute for each BI-enabled segment instance in each structure.

### **Equalization**

You can prevent duplication and the extra workload and complexity that result from the flattening process by assigning a label to any set of segments that serve the same purpose in different structures. BI-enabled segments that have labels are equalized so that the flattened business components include one attribute for each label, consolidated across structures. Non-labeled segments are not equalized across structures, so the flattened components include a separate attribute for each segment for each structure.

---

#### **Note**

It may not be possible to equalize similarly labeled segments if they have incompatible data types or value set types.

---

### **Mapping to Logical Objects in Business Intelligence**

Assign a label to a segment to map the corresponding attribute in the flattened components to a logical object in Oracle Business Intelligence. Using labels to map segments to BI logical objects minimizes the steps for importing the flexfield into Oracle Business Intelligence.

---

#### **Note**

Assigning a label to a segment serves to equalize the attribute across structures, as well as map the equalized attribute to business intelligence.

---

### **Managing Labels**

You may assign a predefined label (if available) to segments or create new labels for assignment, as needed. Specify a code, name, and description to identify each label. In the BI Object Name field, enter the name of the logical object in Oracle Business Intelligence to which the segment label should map during import. Specifying the BI logical object minimizes the steps for importing the flexfield into Oracle Business Intelligence and helps to equalize context-sensitive segments across structures.

If no labels are assigned to a BI-enabled segment, or the BI Object Name on the assigned label does not exist in business intelligence, you must manually map the segment to the desired logical object when importing into Oracle Business Intelligence.

In addition, segments without labels cannot be equalized across structures. The flattened components include a separate attribute for each non-labeled segment in each structure.

---

#### **Note**

Segment labels serve other functions as well, as presented in Key Flexfields: Explained.

---

## **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.

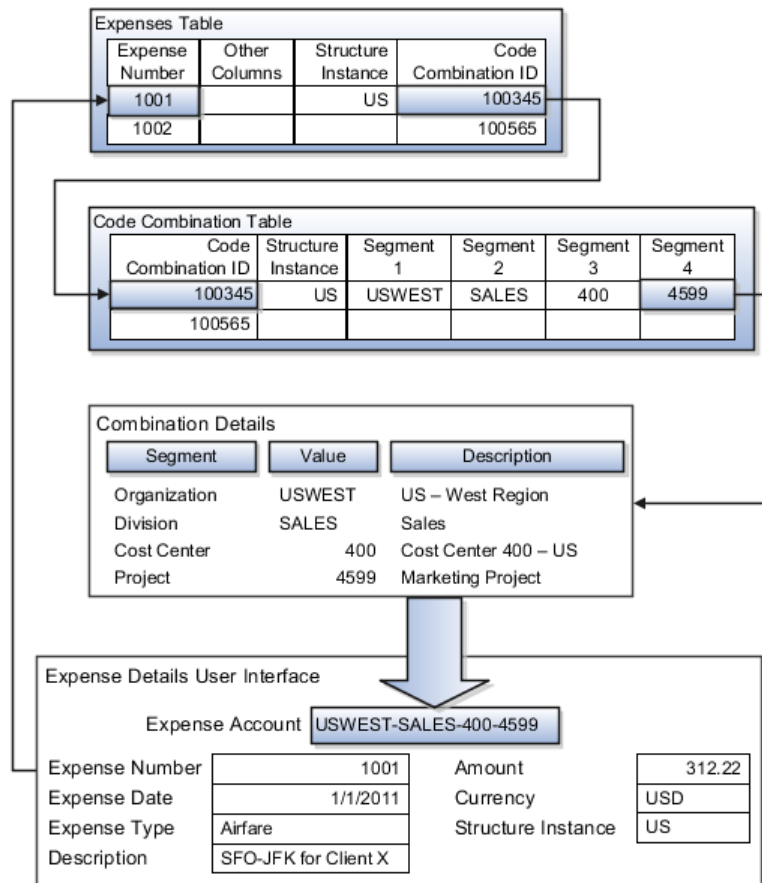
## Analysis

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

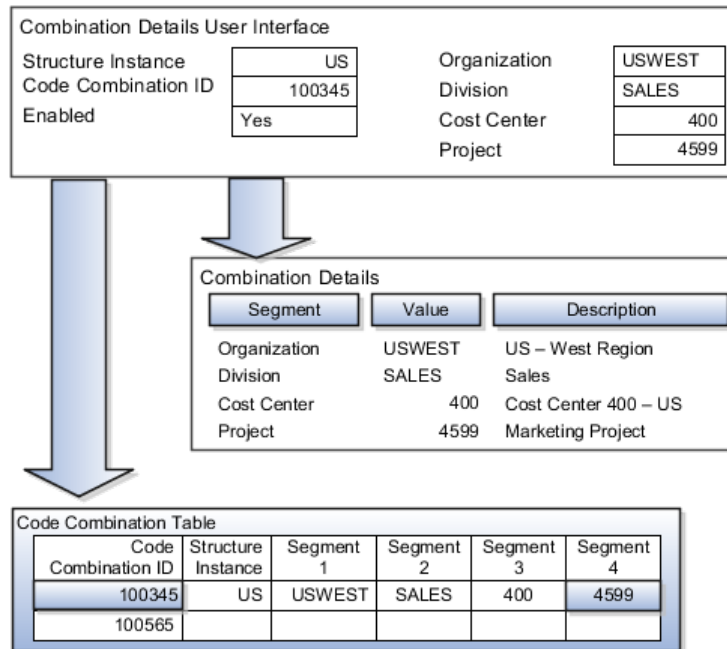
The code combination table supports entering account information, such as for expense accounts.

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.

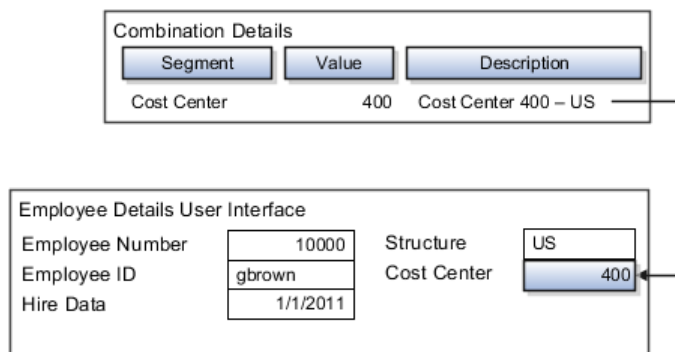


If dynamic combination creation is not enabled, then when entering an expense line, the user can only select an account that already exists in the ACCOUNTS (combinations) table. If they require an account that does not exist, they must consult with the appropriate application administrator who can add 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.

## Define Attachments

### 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.

Action	Privilege
View	Read Application Attachment (FND_READ_APPLICATION_ATTACHMENT_DATA)
Add or Update	Update Application Attachment (FND_UPDATE_APPLICATION_ATTACHMENT_DATA)
Delete	Delete Application Attachment (FND_DELETE_APPLICATION_ATTACHMENT_DATA)

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 Portal: 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.

---

## **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.

---





---

# Define Incentive Compensation Shared Configuration

## Manage Profile Options and Flexfields

### Incentive Compensation Profile Options: Explained

Oracle Fusion Incentive Compensation has two profile options, which you can set at the site level.

- **CN\_DEBUG:** Select **Yes** when you want the application to write the debug errors to the CN\_PROCESS\_AUDIT\_LINES table.
- **CN\_MARK\_EVENTS:** Select **Yes** when you want the application to record every event in the Changed Events Log for inclusion in the next incremental calculation and plan deployer processes.

### Incentive Compensation Flexfields: Explained

All standard, enabled incentive compensation descriptive flexfields relate to transactions. Use them to map transaction attributes in the source application, which do not exist by default in Oracle Fusion Incentive Compensation, to the CN\_TP\_TRANSACTIONS\_ALL table.

The descriptive flexfields consist of two data types: VARCHAR2 and NUMBER

- **VARCHAR2 (240 CHAR):** ATTRIBUTE1 -- ATTRIBUTE150
- **NUMBER:** ATTRIBUTE\_NUMBER1 -- ATTRIBUTE\_NUMBER50

Enable these attributes for use in the Crediting, Classification, and Calculation processes in the Define Business Unit Configuration for Incentive Compensation task list, Configure Tables and Columns task.

## Manage Calendars

### Incentive Compensation Calendars, Periods, and Period Types: How They Work Together

This topic covers how an incentive compensation calendar supports processing activities for incentive compensation business units by first defining period types and periods, and then exploring how they work together.

## Period Type

Determines how you divide your calendar or fiscal year. The following table shows the delivered types and corresponding periods in a year:

Period Type	Periods in a Year
Monthly	12
Quarterly	4
Semi-Annually	2
Yearly	1

Create custom period types as required. Enter the number of periods in a year so that the application can validate the periods that you define for the calendar. Select the shortest required period range (for example, weekly or monthly) to ensure that the application can accumulate attainments, maintain goals and subledger balances, and process payments at this level, or any higher level.

---

### Restriction

- You cannot accumulate attainments for a period type shorter than your selection. For example, if you select **Monthly**, you cannot accumulate attainments weekly.
- You cannot edit or delete standard or custom period types used by any calendar.

---

## Period

Define for each calendar based on the selected period type, including the definition of period with year, period name, sequence, and start and end date--inclusive of both the dates. Abbreviate the period name if you want, for example Jan for January or W1 for Week1. The application arranges periods based on the sequence values that you enter.

---

### Restriction

- The date ranges within one period cannot overlap with other periods and must be continuous with reference to the previous and next periods.
- After you associate a calendar with a business unit and open the first period for the business unit, you cannot edit the calendar.

---

### Tip

Create all of the periods required for the compensation plans, measures, and goals before creating the plans, measures, and goals.

---

## Calendars and Periods: How They Work

The following table shows the various calendar and period tasks that first you and then the application perform.

User Actions	Application Actions
1. Create a calendar and associated periods as a single entity within the application.	Store the calendar data in the calendar entity (CN_CALEDARS) and the period data in the period entity (CN_PERIODS).
2. Assign a calendar to a business unit as part of the business unit set up.	<p>Insert the periods (zero or more) that are assigned to that calendar into the business unit level tables. For each interval type defined for the business unit, insert each period into the CN_CAL_PER_INT_TYPES table.</p> <p>The CN_PERIOD_STATUS table stores basic period data, including the default status (Never Opened) given to each period.</p>
3. Edit the period status (for example, select <b>Open</b> ) within the business unit.	Update the period statuses in the CN_PERIOD_STATUS table accordingly. If required, launch a concurrent request to populate the period data in all of the participant-related (SRP) tables.
4. Create new, future periods in the calendar.	<p>Insert the new period records into the following tables:</p> <ul style="list-style-type: none"> <li>• CN_PERIODS: Track the nonbusiness unit based calendar period</li> <li>• CN_PERIOD_STATUS: Track the periods and associated statuses for each business unit. If there are three business units associated with the calendar, insert the newly created period record into all of the business unit tables.</li> <li>• CN_CAL_PER_INT_TYPES: Group each period into the interval type for each business unit. If there are three business units associated with a calendar, and assuming each one has four interval types (<b>Period</b>, <b>Quarter</b>, <b>Semi-Annual</b>, and <b>Year</b>), insert the newly created period record as 12 records (3 business units * 4 interval types).</li> </ul>

## FAQs for Manage Calendars

### Can I use the same incentive compensation calendar for different business units?

Yes, you can use the **Calendar** choice list in the Define Business Unit Configuration for Incentive Compensation task list, Manage Parameters task to assign the calendar to the business units, if they require the same calendar set up. The application automatically propagates any change to the calendar, such as new periods, to all of the business units using the calendar.

### What happens if I assign an incentive compensation calendar to a business unit?

You can update the calendar name and add more periods, but you cannot change the period type.

### **What happens if I create periods for a new fiscal year in an incentive compensation calendar already used by business units?**

The application adds the new period record to the list of periods maintained in period status and interval related tables, in all the incentive compensation business units to which you assigned the calendar. For example, if you assign a calendar to Vision Operations and Vision Services and you created a Jan-2010 period for this calendar, the application adds this period to the tables maintaining period statuses and interval periods for Vision Operations and Vision Services.

---

#### **Tip**

Create all the periods required for performance measures and goals first.

---

- Correct the interval numbers for the new period under various interval types for the business units.
- Open the period in all of the corresponding business units to use it for transaction processing.

## **Manage Currencies and Currency Conversions**

### **Multicurrency: How Incentive Compensation Uses It**

Oracle Fusion Incentive Compensation enables you to maintain and process transactions in multiple currencies. Support your global workforce by processing the transactions, originating in different source currencies, in a single processing currency or in each participant's home currency. Compensate participants selling in various countries or different business units within one or more incentive compensation business units.

---

#### **Tip**

A given participant can belong to only one business unit. End date the participant in the current business unit and then add the participant to the new one, if you must change the participant's business unit.

---

## **Settings That Affect Multicurrency**

There are currency as well as conversion rate, rate type, and date parameters in the Define Shared Configuration for Incentive Compensation task list, Manage Currencies and Manage Currency Conversions tasks.

- Operating currency
- 

#### **Tip**

Set this value to the currency in which your top management wants to view reports.

---

- Participant home currency

- Processing currency

---

**Tip**

If you set processing currency to:

- **Operating currency**, then the Crediting, Calculation, and Payment processes use the single operating currency for processing.
- **Participant home currency**, then the Crediting, Calculation, and Payment processes use each participant's home currency for processing. Also, you may have to individualize participant compensation plan details, such as rates and goals, because you establish compensation plan elements at the business unit level.

---

**Note**

The application maintains participant subledger balances only in the processing currency. A participant may have subledger balances in more than one currency if you set the processing currency as **Participant home currency**, and the participant's home currency changes.

---

**Important**

Reassign incentive plans if you change the participant's currency.

Choose which rate to use to convert amounts among the four currency options. Create rate types such as Corporate, Daily, and Period Average, and define rates differently under each one of them to use in different scenarios. For example, you decide to use Daily rates to convert transaction and credit amounts, and Period Average rates to convert payment amounts. In this case, use these two parameters to set appropriate rate types for these processes.

- **Payment Conversion Rate Type**: The application uses the rates within the selected rate type to convert the payment amount from processing currency to operating currency or home currency.
- **Transaction Conversion Rate Type**: The application uses the rates within the selected rate type for conversion during the Collection, Crediting, and Calculation processes as well as while adjusting transactions.

The **Payment Conversion Date** parameter determines whether the application uses the transaction date or the payment batch pay date to look up the corresponding currency rate to use for payment conversion.

## How Currency Is Stored

To support multicurrency processing, the application stores transactions, credits, earnings, and payments in the following currencies.

Entity	Source Currency	Processing Currency	Operating Currency	Participant Home Currency
Transactions	Yes		Yes	

Credits	Yes	Yes*	Yes	Yes
Earnings		Yes*	Yes	Yes
Payments		Yes*	Yes	Yes

\* Represents the core currency associated with the transactional entity. Other currencies are available for transaction traceability and reporting. For example, when you adjust the amount of a base transaction, the application does the adjustment in the source currency, whereas, by default, the application shows the credit and earning amounts in the processing currency.

---

#### Tip

The application maintains attainments only in processing currency.

---

### Processing in Participant Home Currency Example

Assume you are a compensation manager working for multinational company ABC with a global sales force. You decided to compensate North American salespersons working in Canada and the United States in a single incentive compensation business unit with the operating currency of **USD**. The processing currency is **Participant Home Currency** so you compensate the Canadian sales force in its home currency of CAD and the US sales force in USD.

Collection: Since your salespeople are selling in the US and Canada, you collect transactions in USD and CAD, into the same business unit. Now, you track two transactions, one booked in US dollars and another in Canadian dollars, assuming an exchange rate of 1 CAD equals 0.9302326 USD.

---

#### Note

The application converted the transaction amount from source currency to operating currency and stores it for each transaction. It does any adjustment to the transaction using the source currency amount.

---

Transaction Number	Transaction Event Date	Transaction Amount in Source Currency	Transaction Amount in Operating Currency
T01012010-1	1-Jan-2010	10,000 USD	10,000 USD
T01022010-9	1-Feb-2010	21,500 CAD	20,000 USD

Crediting: The application manages credit and roll up creation in source currency and adjustments in the processing currency. It automatically converts and populates the source currency, operating currency, and home currency credit amount in all of the previous cases. Assume that the two orders are for a team of two salespeople, one working from the US and another from Canada, sharing 50 percent credit. The application generates credits in the home currency of the credit receiver because the processing currency is set to participant home currency.

Transaction Number	Transaction Event Date	Credit Receiver	Credit Amount in	Credit Amount	Credit Amount in	Credit Amount
--------------------	------------------------	-----------------	------------------	---------------	------------------	---------------

			Processing Currency	in Source Currency	Operating Currency	in Home Currency
T01012010-1	1-Jan-2010	US Salesperson	5,000 USD	5,000 USD	5,000 USD	5,000 USD
T01012010-1	1-Jan-2010	Canadian Salesperson	5,375 CAD	5,000 USD	5,000 USD	5,375 CAD
T01022010-9	1-Feb-2010	US Salesperson	10,000 USD	10,750 CAD	10,000 USD	10,000 USD
T01022010-9	1-Feb-2010	Canadian Salesperson	10,750 CAD	10,750 CAD	10,000 USD	10,750 CAD

Calculation: The application calculates the earnings in the processing currency and also converts them to, and stores them in, operating currency and home currency. Assume that both salespeople are on the same plan with individualized rates and goals and that they get earnings (10 percent of credited amount) as follows:

Transaction Number	Transaction Event Date	Credit Receiver	Earning Amount in Processing Currency	Earning Amount in Operating Currency	Earning Amount in Home Currency
T01012010-1	1-Jan-2010	US Salesperson	500 USD	500 USD	500 USD
T01012010-1	1-Jan-2010	Canadian Salesperson	537.5 CAD	500 USD	537.5 CAD
T01022010-9	1-Feb-2010	US Salesperson	1,000 USD	1,000 USD	1,000 USD
T01022010-9	1-Feb-2010	Canadian Salesperson	1,075 CAD	1,000 USD	1,075 CAD

---

#### Note

The three previous processes use the currency conversion rates defined under the transaction conversion rate type parameter.

---

Payment: The application computes the payment amounts in the processing currency and also converts them to home currency and operating currency. It does payment and manual adjustments in processing currency and again converts and stores them in home currency and operating currency.

Transaction Number	Transaction Event Date	Credit Receiver	Payment Amount in Processing Currency	Payment Amount in Operating Currency	Payment Amount in Home Currency
T01012010-1	1-Jan-2010	US Salesperson	500 USD	500 USD	500 USD
T01012010-1	1-Jan-2010	Canadian Salesperson	537.5 CAD	500 USD	537.5 CAD
T01022010-9	1-Feb-2010	US Salesperson	1,000 USD	1,000 USD	1,000 USD
T01022010-9	1-Feb-2010	Canadian Salesperson	1,075 CAD	1,000 USD	1,075 CAD

---

#### Note

The Payment process uses the currency conversion rates defined under the payment conversion rate type. The rate that the application uses for conversion is based on the Payment Conversion Date parameter.

### Processing in Operating Currency Example

This example uses the data from the previous example to show how the application processes and stores transactions when you set operating currency to **Processing Currency**. The changes are shown in processing currency columns in the following tables.

#### Collection

Transaction Number	Transaction Event Date	Transaction Amount in Source Currency	Transaction Amount in Operating Currency
T01012010-1	1-Jan-2010	10,000 USD	10,000 USD
T01022010-9	1-Feb-2010	21,500 CAD	20,000 USD

#### Crediting

Transaction Number	Transaction Event Date	Credit Receiver	Credit Amount in Processing Currency	Credit Amount in Source Currency	Credit Amount in Operating Currency	Credit Amount in Home Currency
T01012010-1	1-Jan-2010	US Salesperson	5,000 USD	5,000 USD	5,000 USD	5,000 USD
T01012010-1	1-Jan-2010	Canadian Salesperson	5,000 USD	5,000 USD	5,000 USD	5,375 CAD
T01022010-9	1-Feb-2010	US Salesperson	10,000 USD	10,750 CAD	10,000 USD	10,000 USD
T01022010-9	1-Feb-2010	Canadian Salesperson	10,000 USD	10,750 CAD	10,000 USD	10,750 CAD

#### Calculation

Transaction Number	Transaction Event Date	Credit Receiver	Earning Amount in Processing Currency	Earning Amount in Operating Currency	Earning Amount in Home Currency
T01012010-1	1-Jan-2010	US Salesperson	500 USD	500 USD	500 USD
T01012010-1	1-Jan-2010	Canadian Salesperson	500 USD	500 USD	537.5 CAD
T01022010-9	1-Feb-2010	US Salesperson	1,000 USD	1,000 USD	1,000 USD
T01022010-9	1-Feb-2010	Canadian Salesperson	1,000 USD	1,000 USD	1,075 CAD

#### Payment

Transaction Number	Transaction Event Date	Credit Receiver	Payment Amount in	Payment Amount in	Payment Amount in Home Currency
--------------------	------------------------	-----------------	-------------------	-------------------	---------------------------------



			Processing Currency	Operating Currency	
T01012010-1	1-Jan-2010	US Salesperson	5,00 USD	500 USD	500 USD
T01012010-1	1-Jan-2010	Canadian Salesperson	500 USD	500 USD	537.5 CAD
T01022010-9	1-Feb-2010	US Salesperson	1,000 USD	1,000 USD	1,000 USD
T01022010-9	1-Feb-2010	Canadian Salesperson	1,000 USD	1,000 USD	1,075 CAD

## Incentive Compensation Currencies: How They Are Converted

The application maintains incentive compensation currency conversions for instances where the sale currency is different from the operating currency or the participant home currency as well as when you want to use a different conversion rate than the default daily rates, for example, if you want to use period-based rates.

## Settings That Affect Currency Conversion

In the Define Shared Configuration for Incentive Compensation task list, Manage Currency Conversions task, create your rate types and associated rates. Then, use the Define Business Unit Configuration for Incentive Compensation task list, Manage Parameters task to make your selections for these parameters:

- **Processing Currency:** Determines how the application actually converts currencies during processing. There is a Calculation Currency column on each transaction entity, which holds the value of the processing currency. Processing currency is either **Operating currency** (in which case every transaction would record the same currency) or **Participant home currency**.

---

### Important

After you run the Calculation process or close a period, you cannot edit the value for this parameter.

---

- **Transaction Conversion Rate Type**
- **Collection Conversion Rate Type**
- **Payment Conversion Rate Type**

## How Currency Is Converted

1. Select one of the standard conversion rate types or create a custom one.

Create custom rate types (using unique names) based on your business requirements. For example, if your business units use period average

based rates for conversion, create a new rate type, enter start and end dates (based on your calendar period definition), and provide the period-average rates for the date range.

## 2. Create or edit conversion rates.

Enter the required values. If you also enter an end date, the start date must be earlier than the end date for each record. Create two records for each date range and from and to currencies where the second record is the inversion of the first. For example, you create one record with an exchange rate from USD to CAD for the start date 1-Jun-2010 and a second record with an exchange rate from CAD to USD for the same date. The application may use the inverse rate for reference purposes to compute the transactional amount from the operating amount, if required.

---

### Restriction

For the from and to currencies and date range for a rate type, you can use only one rate. If you want different rates for the same date range and set of from and to currencies, define a different rate type.

---

### Restriction

You can edit rates as long as the incentive compensation period is not closed in any of the associated incentive compensation business units.

---

## FAQs for Manage Currency Conversions

### How can I use different incentive compensation rates to process transactions and payments in different incentive compensation business units?

Define multiple incentive compensation currency rate types for the same date, different incentive compensation business units, or different usage and associate the appropriate rate type with each business unit. The rate types may match those in your general ledger, but it is not required. For example, one business unit requires daily-based rates where the rate is different for each day. Another business unit requires period-based rates where the period is monthly and each period has the same rate, which might be the average or end rate for the period. Yet another business unit also requires period-based rates, but its period is weekly instead of monthly.

### Where does incentive compensation use the transaction and payment conversion rate types?

The application uses the currency conversion rates that you defined for the rate type (in the Define Shared Configuration for Incentive Compensation task list, Manage Currency Conversions task) , and selected for **Transaction Conversion Rate Type** and **Payment Conversion Rate Type** (in the Define Business Unit Configuration for Incentive Compensation task list, Manage Parameters task), to convert transaction and payment amounts from one currency to another during Crediting, Rollup, and Calculation processing.

The application stores transactions, credits, earnings, and payments in various currencies. It uses the rates to convert currencies among the various amounts shown here.

Amounts in Entities	Source Currency	Processing Currency	Operating Currency	Participant Home Currency
Transactions	Yes		Yes	
Credits	Yes	Yes	Yes	Yes
Earnings		Yes	Yes	Yes
Payments		Yes	Yes	Yes



---

## Define Incentive Compensation Custom Qualifiers and Lookups

### Manage Incentive Compensation Custom Qualifiers and Lookups: Explained

The Manage Incentive Compensation Custom Qualifiers and Lookups task list enables you to more easily create incentive compensation custom rule qualifiers for use in crediting and classification rules.

The tasks that comprise the Manage Incentive Compensation Custom Qualifiers and Lookups task list are:

1. Register Custom Qualifier Tables and Views
2. Manage Incentive Compensation Lookups
3. Manage Value Sets for Custom Qualifiers
4. Manage Descriptive Flexfields for Custom Qualifiers
5. Manage Custom Rule Qualifiers

Access this task list and associated tasks from the Setup and Maintenance work area.

#### Register Custom Qualifier Tables and Views

Registered tables, views, and lookups are used to dynamically create the **Business Object** choice list on the Create Value Set for Custom Qualifier page. Objects that are not registered are not available in that choice list and cannot be used to create a value set. You can register custom objects as well as any lookups where the selected module is **Incentive Compensation**.

#### Manage Incentive Compensation Lookups

To create a new incentive compensation lookup, start the lookup type with **CN\_** and set the module to **Incentive Compensation**.

---

#### Restriction

You can only register custom qualifier lookups when the selected module is **Incentive Compensation**.

---

## Manage Value Sets for Custom Qualifiers

On the Create Value Set for Custom Qualifier page, you select the business object from a choice list that contains only the tables, views, and lookups that were registered using the Register Custom Qualifiers Tables and Views page. You can also validate whether the value set satisfies your business requirements before creating the custom rule qualifier. To perform the validation, add a data filter for the new value set and run a test to see up to the top 100 results for the display column and column ID or column values for the value set.

---

### Tip

To create a value set using lookups where the selected module is not **Incentive Compensation**, use the common Manage Value Sets task. Then, associate that value set with the descriptive flexfield segment for the custom qualifier.

---

## Manage Descriptive Flexfields for Custom Qualifiers

When you create a global segment, it is recommended that you also select **BI Enabled** so that the attribute is available for use in business intelligence dashboards and reports.

## Manage Custom Rule Qualifiers

This task expose the attribute on the relevant Oracle Fusion Incentive Compensation rule and transaction pages, as well as any transaction spreadsheets that you generate after completing the task.

---

## Define Incentive Compensation Business Unit Configuration

### Manage Intervals

#### Incentive Compensation Interval Types: How They Work

The application uses interval types to group specific calendar periods to accumulate achievements (for example interval-to-date attainment), earnings (for example, year-to-date earning), or payments during Calculation or Payment processing. There are four defined interval types, **Year**, **Semi-Annual**, **Quarter**, and **Period** and you can create custom ones, as required. The **Period** interval type has the same definition as the calendar period.

---

#### Tip

Group periods into an interval by giving the same interval number to each period to include in the group.

---

#### Creating Quarterly Intervals: Example

This is an example of how to use interval numbers to group periods into quarters.

Period	Year	From Date	To Date	Interval Number
Jan-10	2010	1-Jan-2010	31-Jan-2010	2010001
Feb-10	2010	1-Feb-2010	28-Feb-2010	2010001
Mar-10	2010	1-Mar-2010	31-Mar-2010	2010001
Apr-10	2010	1-Apr-2010	30-Apr-2010	2010002
May-10	2010	1-May-2010	31-May-2010	2010002
Jun-10	2010	1-Jun-2010	30-Jun-2010	2010002
Jul-10	2010	1-Jul-2010	31-Jul-2010	2010003
Aug-10	2010	1-Aug-2010	31-Aug-2010	2010003
Sep-10	2010	1-Sep-2010	30-Sep-2010	2010003
Oct-10	2010	1-Oct-2010	31-Oct-2010	2010004

Nov-10	2010	1-Nov-2010	30-Nov-2010	2010004
Dec-10	2010	1-Dec-2010	31-Dec-2010	2010004

### Creating Yearly Intervals: Example

This is an example of how to use an interval number to group periods into a year.

Period	Year	From Date	To Date	Interval Number
Jan-10	2010	1-Jan-2010	31-Jan-2010	2010
Feb-10	2010	1-Feb-2010	28-Feb-2010	2010
Mar-10	2010	1-Mar-2010	31-Mar-2010	2010
Apr-10	2010	1-Apr-2010	30-Apr-2010	2010
May-10	2010	1-May-2010	31-May-2010	2010
Jun-10	2010	1-Jun-2010	30-Jun-2010	2010
Jul-10	2010	1-Jul-2010	31-Jul-2010	2010
Aug-10	2010	1-Aug-2010	31-Aug-2010	2010
Sep-10	2010	1-Sep-2010	30-Sep-2010	2010
Oct-10	2010	1-Oct-2010	31-Oct-2010	2010
Nov-10	2010	1-Nov-2010	30-Nov-2010	2010
Dec-10	2010	1-Dec-2010	31-Dec-2010	2010

## FAQs for Manage Intervals

### What happens if I edit interval numbers after using them in incentive compensation processing?

Run the Calculation and Payment processes again for all of the periods affected by the change in the interval numbers.

### Where do incentive compensation interval types get used?

The application uses them in performance measures as performance intervals to accumulate attainment information and as goal intervals. It also uses them in plan components to specify how frequently to calculate earnings (payout frequency) and in incentive payment plans as payment intervals to define draw and cap rules.

You cannot delete a custom interval type if it is used in any of the previous manners.

### Why can't I delete an incentive compensation interval type?

You cannot delete the delivered incentive compensation interval types Year, Semi-Annual, Quarter, and Period nor can you delete any custom interval type that you created if anyone used it in a performance measure, plan component, goal, or payment plan.



## Configure Tables and Columns

### Enabling an Incentive Compensation Attribute as a Qualifier for Classification or Crediting Rules: Worked Example

This example demonstrates how to create a custom incentive compensation attribute with a list of values for use in crediting rules for those situations where the delivered attributes do not fulfill business requirements. The application has these attributes (in the CN\_TP\_TRANSACTIONS\_ALL table) enabled for use in crediting and classification rules:

- AREA\_CODE
- CITY
- COUNTRY
- CUSTOMER\_ID

---

**Tip**

Use the customer (directory id) or customer name range for matching.

- 
- INVENTORY\_ITEM\_ID

---

**Tip**

Use the product id or sales catalog hierarchy for matching.

- 
- ORG\_ID
  - PARTICIPANT\_ID
  - POSTAL\_CODE
  - PROVINCE
  - ROLE\_ID (enabled for crediting rules only)
  - SALES\_CHANNEL
  - SOURCE\_ORG\_ID
  - STATE
  - TERR\_NAME
  - TRANSACTION\_AMT\_FUNC\_CURR
  - TRANSACTION\_QTY
  - TRANSACTION\_TYPE

The three tasks covered in this topic are:

- Run SQL script to enable a custom attribute as a qualifier for crediting rules.
- Enable the custom attribute.
- Edit and deploy the relevant descriptive flexfield.

---

**Tip**

Also use the steps in these tasks to create a custom attribute for classification rules by changing the usage\_id from -1001 to -1002.

---

## Running Script to Enable Custom Attribute

In this task, use SQL and a tool such as SQL Developer to create new FND LOOKUP values to use in the choice list on the user interface (optional), run the CN\_RS\_CUST\_ATTR.CREATE\_QUAL application programming interface (API) to create the custom attribute, and update the BATCH\_TOTAL\_SQL to assign the new custom attribute.

1. Create new FND LOOKUP values to use in the choice list on the user interface (optional).

```
-- Example: create a Line of Business lookup
-- LOOKUP_TYPE = CN_LINE_OF_BUSINESS
DECLARE
    l_retcode VARCHAR2(30);
    l_errbuf VARCHAR2(10000);
BEGIN
    -- create the FND lookup TYPE
    fnd_lookup_types_pkg.CREATE_OR_UPDATE_ROW(
        X_VIEW_APPSNAME => 'FND',
        X_LOOKUP_TYPE => 'CN_LINE_OF_BUSINESS',
        X_APPLICATION_SHORT_NAME => 'CN',
        X_MEANING => 'Line of Business',
        X_DESCRIPTION => 'Fusion Example: Line of Business',
        X_REFERENCE_GROUP_NAME => '');
    -- populate the FND lookup values
    fnd_lookup_values_pkg.CREATE_OR_UPDATE_ROW(
        X_LOOKUP_TYPE => 'CN_LINE_OF_BUSINESS',
        X_VIEW_APPSNAME => 'FND',
        X_LOOKUP_CODE => 'COMMERCIAL_LEASE',
        X_MEANING => 'Commercial Leasing',
        X_SET_CODE => '',
        X_DESCRIPTION => 'Commercial Leasing',
        X_ENABLED_FLAG => 'Y',
        X_START_DATE_ACTIVE => sysdate,
        X_END_DATE_ACTIVE => '');
    fnd_lookup_values_pkg.CREATE_OR_UPDATE_ROW(
        X_LOOKUP_TYPE => 'CN_LINE_OF_BUSINESS',
        X_VIEW_APPSNAME => 'FND',
        X_LOOKUP_CODE => 'RESIDENTIAL',
        X_MEANING => 'Residential',
        X_SET_CODE => '',
        X_DESCRIPTION => 'Residential',
        X_ENABLED_FLAG => 'Y',
        X_START_DATE_ACTIVE => sysdate,
        X_END_DATE_ACTIVE => '');
    fnd_lookup_values_pkg.CREATE_OR_UPDATE_ROW(
        X_LOOKUP_TYPE => 'CN_LINE_OF_BUSINESS',
        X_VIEW_APPSNAME => 'FND',
        X_LOOKUP_CODE => 'CONSUMER_LOAN',
        X_MEANING => 'Consumer Personal Loans',
        X_SET_CODE => '',
        X_DESCRIPTION => 'Consumer Personal Loans',
        X_ENABLED_FLAG => 'Y',
        X_START_DATE_ACTIVE => sysdate,
        X_END_DATE_ACTIVE => '');
```

END;  
/

2. Run the CN\_RS\_CUST\_ATTR.CREATE\_QUAL application programming interface (API) to create the custom attribute.

API Parameter	Description	Sample Value
p_attribute_id	Get the current minimum attribut_id and decrement it by 1  sql> select min(attribute_id) -1 from cn_rs_attributes_all_	-2026
p_object_version_number	Not available	1
p_org_id	sql> select organization_id, name from hr_all_organization_u where name like 'Vision Services%' ;	458
p_source_lang	Not available	'US'
p_language	Not available	'US'
p_name	This is the user interface display name and you can enter any name	'Line of Business'
p_description	Not available	'Custom Attribute : Line of Business'
p_attr_usage_id	Use any number or pick the next number from here:  sql> select min(attr_usage_id) from cn_rs_attr_usages_all WHERE usage_id = -1001  It is good to use the same value p_attribute_id, unless it is already assigned.	-2026
p_usage_id	usage_id = -1001 for crediting  usage_id = -1002 for classification	-1001

p_indexed_col_name	<p>Corresponds to cn_rs_usage_details_all table batch_total_sql. Use the next available value for the same data type, in the unused column. To see which indexed_col_name are used:</p> <pre>sql&gt; select     enabled_flag,     batch_total_sql from     CN_RS_USAGE_DETAILS_A where usage_id =     -1001 and org_id = 458 ;</pre>	'SQUAL_CHAR20'
p_display_type	<p>'CHAR' = varchar or character</p> <p>'NUMERIC' = number</p>	'CHAR'
p_source_column_name	<p>From the batch_total_sql field, there are two tables CN_TP_TRANSACTIONS and HZ_PARTIES. Use one of these fields as the source_column_name. If you want to use a field from other tables, then batch_total_sql must include those tables.</p> <pre>sql&gt; desc     fusion.CN_TP_TRANSACT sql&gt; desc     fusion.HZ_PARTIES</pre>	'ATTRIBUTE38'

p_attr_relation_factor	<p>Get the next prime number, see <a href="http://primes.utm.edu/lists/small/1000.txt">http://primes.utm.edu/lists/small/1000.txt</a>. To see which prime number is already used:</p> <pre>sql&gt; select   attr_relation_factor,   indexed_col_name,   attribute_id from   CN_RS_ATTR_USAGES_ALL where usage_id =   -1001 and org_id = 458 order by   attr_relation_factor desc ;</pre>	227
p_comparison_operator_cn	<p>Define the value for this custom attribute. Use the positive value of p_attribute_id.</p> <p>Naming convention: q&lt;p_attribute_id&gt;_cp</p>	'q2026_cp'
p_low_value_char_cn	<p>Populate if using the <b>equal</b> or <b>between</b> parameter.</p> <p>Naming convention: q&lt;p_attribute_id&gt;_low_va</p>	'q2026_low_value_char'
p_high_value_char_cn	<p>Populate if using the <b>between</b> parameter.</p> <p>Naming convention: q&lt;p_attr_usage_id&gt;_high</p>	'q2026_high_value_char'
p_equal_flag	Enable the EQUAL operator	'Y'
p_batch_op_eol	SQL syntax for EQUAL operand if p_equal_flag = 'Y'	'( A.squal_char20 = B.q2026_low_value_char AND B.q2026_cp = "=" )'
p_like_flag	Enable the LIKE operator	'Y'
p_batch_op_like	SQL syntax for EQUAL operand if P_LIKE_FLAG = 'Y'	'( A.squal_char20 like B.q2026_low_value_char AND B.q2026_cp = "LIKE" )'
p_between_flag	Enable the BETWEEN operator	'Y'

p_batch_op_between	SQL syntax for EQUAL operand if P_BETWEEN_FLAG = 'Y'	'( A.squal_char20 between B.q2026_low_value_char and B.q2026_high_value_char AND B.q2026_cp = "BETWEEN" )'
p_convert_to_id_flag	Convert the varchar2 column to id column, set this value to 'Y'	'N'
p_low_value_char_id_cn	Convert the varchar2 column to id column. Only edit if this P_CONVERT_TO_ID_FLAG = 'Y'	null
p_html_lov_sql1	If you want an list of value (LOV) to be associated to the attribute.	'select /*+ FIRST_ROWS(10) */ 1 lookup_id, meaning meaning, lookup_code lookup_code from cn_lookups where lookup_type = "CN_EXAMPLE_LINE_OF_BUSINESS" ,

```
-- Example: ATTRIBUTE38 is enabled for crediting with list of value
-- from CN_LINE_OF_BUSINESS FND lookup
```

```
DECLARE
```

```
l_retcode VARCHAR2(30);
```

```
l_errbuf VARCHAR2(10000);
```

```
BEGIN
```

```
CN_RS_CUST_ATTR.CREATE_QUAL(
```

```
  p_attribute_id => -2026,
```

```
  p_object_version_number => 1,
```

```
  p_org_id => 458,
```

```
  p_SOURCE_LANG => 'US',
```

```
  p_LANGUAGE => 'US',
```

```
  p_name => 'Line of Business',
```

```
  p_description => 'Custom Attribute : Line of Business',
```

```
  p_ATTR_USAGE_ID => -2026,
```

```
  p_USAGE_ID => -1001,
```

```
  p_INDEXED_COL_NAME => 'SQUAL_CHAR20',
```

```
  p_DISPLAY_TYPE => 'CHAR',
```

```
  p_SOURCE_COLUMN_NAME => 'ATTRIBUTE38',
```

```
  p_ATTR_RELATION_FACTOR => 227,
```

```
  p_CONVERT_TO_ID_FLAG => 'N',
```

```
  p_COMPARISON_OPERATOR_CN => 'q2026_cp',
```

```
  p_LOW_VALUE_CHAR_CN => 'q2026_low_value_char',
```

```
  p_HIGH_VALUE_CHAR_CN => NULL,
```

```
  p_LOW_VALUE_CHAR_ID_CN => NULL,
```

```
  p_LOW_VALUE_NUMBER_CN => NULL,
```

```

p_HIGH_VALUE_NUMBER_CN => NULL,
p_VALUE1_ID_CN => NULL,
p_VALUE2_ID_CN => NULL,
p_VALUE3_ID_CN => NULL,
p_VALUE4_ID_CN => NULL,
p_FIRST_CHAR_CN => NULL,
p_CURRENCY_CODE_CN => NULL,
p_EQUAL_FLAG => 'Y',
p_BATCH_OP_EQL => '( A.squal_char20 = B.q2026_low_value_char AND
B.q2026_cp = ''='' )',
p_LIKE_FLAG => 'N',
p_BATCH_OP_LIKE => NULL,
p_BETWEEN_FLAG => 'N',
p_BATCH_OP_BETWEEN => NULL,
p_BATCH_OP_COMMON_WHERE => NULL,
p_SEEDED_FLAG => NULL,
p_ALIAS_RULE1 => NULL,
p_REAL_TIME_SELECT => NULL,
p_REAL_TIME_WHERE => NULL,
p_REAL_TIME_FROM => NULL,
p_HTML_LOV_SQL1 => 'select /*+ FIRST_ROWS(10) */ 1 lookup_id, meaning
meaning, lookup_code lookup_code from cn_lookups where lookup_type =
''CN_EXAMPLE_LINE_OF_BUSINESS'' ',
p_HTML_LOV_SQL2 => NULL,
p_HTML_LOV_SQL3 => NULL,
p_HIERARCHY_TYPE => NULL,
retcode => l_retcode,
errbuf => l_errbuf);
END;
/

```

### 3. Update the BATCH\_TOTAL\_SQL to assign the new custom attribute.

---

#### Important

Run this SQL to see which fields are already enabled in the existing batch\_total\_sql's select statement.

```

sql> select batch_total_sql from cn_rs_usage_details_all where usage_id =
-1001 and org_id = &org_id ;

```

---

```

DECLARE
l_retcode VARCHAR2(30);
l_errbuf VARCHAR2(10000);
BEGIN
update cn_rs_usage_details_all
set batch_total_sql =
' SELECT T.TRANSACTION_ID TRANS_OBJECT_ID , ' ||
' T.PARTICIPANT_ID ASSIGNED_OBJECT_ID , ' ||
' T.SOURCE_EVENT_DATE EVENT_DATE , ' ||
' T.WORKER_ID , ' ||
' UPPER(substr(C.ACCOUNT_NAME,1,1)) SQUAL_FC01 , ' ||
' null SQUAL_CURC01 , ' ||
' T.SALES_CHANNEL SQUAL_CHAR01 , ' ||

```

```

' T.CITY SQUAL_CHAR02 , ' ||
' T.STATE SQUAL_CHAR03 , ' ||
' T.COUNTRY SQUAL_CHAR04 , ' ||
' C.ACCOUNT_NAME SQUAL_CHAR05 , ' ||
' P.PARTY_NUMBER SQUAL_CHAR06 , ' ||
' P.CATEGORY_CODE SQUAL_CHAR07 , ' ||
' T.POSTAL_CODE SQUAL_CHAR08 , ' ||
' T.PROVINCE SQUAL_CHAR09 , ' ||
' T.AREA_CODE SQUAL_CHAR10 , ' ||
' T.CUSTOMER_ID SQUAL_NUM01 , ' ||
' T.INVENTORY_ITEM_ID SQUAL_NUM02 , ' ||
' NULL SQUAL_NUM03 , ' ||
' T.SOURCE_ORG_ID SQUAL_NUM04 , ' ||
' T.PARTICIPANT_ID SQUAL_NUM05 , ' ||
' null SQUAL_FC02 , ' ||
' null SQUAL_CURC02 , ' ||
' NULL SQUAL_CHAR11 , ' ||
' T.TRANSACTION_TYPE SQUAL_CHAR12 , ' ||
' T.TERR_NAME SQUAL_CHAR13 , ' ||
' NULL SQUAL_CHAR14 , ' ||
' NULL SQUAL_CHAR15 , ' ||
' NULL SQUAL_CHAR16 , ' ||
' NULL SQUAL_CHAR17 , ' ||
' NULL SQUAL_CHAR18 , ' ||
' null SQUAL_CHAR19 , ' ||
' T.ATTRIBUTE38 SQUAL_CHAR20 , ' ||
' T.TRANSACTION_AMT_FUNC_CURR SQUAL_NUM06 , ' ||
' T.TRANSACTION_QTY SQUAL_NUM07 , ' ||
' T.ORG_ID SQUAL_NUM08 , ' ||
' null SQUAL_NUM09 , ' ||
' null SQUAL_NUM10 , ' ||
' null SQUAL_FC03 , ' ||
' null SQUAL_CURC03 , ' ||
' null SQUAL_CHAR21 , ' ||
' null SQUAL_CHAR22 , ' ||
' null SQUAL_CHAR23 , ' ||
' null SQUAL_CHAR24 , ' ||
' null SQUAL_CHAR25 , ' ||
' null SQUAL_CHAR26 , ' ||
' null SQUAL_CHAR27 , ' ||
' null SQUAL_CHAR28 , ' ||
' null SQUAL_CHAR29 , ' ||
' null SQUAL_CHAR30 , ' ||
' null SQUAL_NUM11 , ' ||
' null SQUAL_NUM12 , ' ||
' null SQUAL_NUM13 , ' ||
' null SQUAL_NUM14 , ' ||
' null SQUAL_NUM15 , ' ||
' null SQUAL_FC04 , ' ||
' null SQUAL_CURC04 , ' ||
' null SQUAL_CHAR31 , ' ||
' null SQUAL_CHAR32 , ' ||
' null SQUAL_CHAR33 , ' ||
' null SQUAL_CHAR34 , ' ||

```



```

' null SQUAL_CHAR35 , ' ||
' null SQUAL_CHAR36 , ' ||
' null SQUAL_CHAR37 , ' ||
' null SQUAL_CHAR38 , ' ||
' null SQUAL_CHAR39 , ' ||
' null SQUAL_CHAR40 , ' ||
' null SQUAL_NUM16 , ' ||
' null SQUAL_NUM17 , ' ||
' null SQUAL_NUM18 , ' ||
' null SQUAL_NUM19 , ' ||
' null SQUAL_NUM20 , ' ||
' null SQUAL_FC05 , ' ||
' null SQUAL_CURC05 , ' ||
' null SQUAL_CHAR41 , ' ||
' null SQUAL_CHAR42 , ' ||
' null SQUAL_CHAR43 , ' ||
' null SQUAL_CHAR44 , ' ||
' null SQUAL_CHAR45 , ' ||
' null SQUAL_CHAR46 , ' ||
' null SQUAL_CHAR47 , ' ||
' null SQUAL_CHAR48 , ' ||
' null SQUAL_CHAR49 , ' ||
' null SQUAL_CHAR50 , ' ||
' null SQUAL_NUM21 , ' ||
' null SQUAL_NUM22 , ' ||
' null SQUAL_NUM23 , ' ||
' null SQUAL_NUM24 , ' ||
' null SQUAL_NUM25 ' ||
' from CN_TP_TRANSACTIONS_ALL T , ' ||
' HZ_PARTIES P ' ||
' where T.CUSTOMER_ID = P.PARTY_ID(+) '
where usage_id = -1001;
END;
/

```

## Enabling Custom Attribute for Crediting

1. Go to the Define Business Unit Configuration for Incentive Compensation task list, Configure Tables and Columns page.
2. In the Tables section, select **CN\_TP\_TRANSACTIONS\_ALL**.
3. Scroll to your attribute and select **Enable for Crediting**.

If you want to enable the attribute for classification, then select **Enable for Classification**.

4. Click **Save and Close**.

---

### Tip

To expose your customer attribute in the user interface, and in application-generated spreadsheets, edit and deploy the descriptive flexfield.

---

## Adding Incentive Compensation Varchar and Date Custom Qualifiers to Crediting Rules: Examples

This topic contains two sample SQL scripts for adding incentive compensation crediting rules qualifiers that correspond to the flexfield Attribute1 and Invoice\_Date in the CN\_TP\_TRANSACTIONS\_ALL table. The script includes comments to help you customize the SQL to your business requirements.

### Adding an Incentive Compensation Varchar Qualifier to Crediting Rules

```
declare
  l_retcode VARCHAR2(30);
  l_errbuf VARCHAR2(10000);

begin
  CN_RS_CUST_ATTR.CREATE_QUAL(
    p_attribute_id => -1026, -- unique number in the cn_rs_attributes_all
    table
    p_object_version_number => 1,
    p_org_id => 204, -- operating unit
    p_SOURCE_LANG => 'US',
    p_LANGUAGE => 'US',
    p_name => 'Attribute 1', -- display name
    p_description => 'Custom Attribute: Attribute1',
    p_ATTR_USAGE_ID => -1026, -- unique number in cn_rs_attr_usages_all
    p_USAGE_ID => -1001,
    p_INDEXED_COL_NAME => 'SQUAL_CHAR15', --corresponds to
    cn_rs_usage_details_all table batch_total_sql
    p_DISPLAY_TYPE => 'CHAR',
    p_SOURCE_COLUMN_NAME => 'ATTRIBUTE1', -- corresponds to
    cn_tp_transactions_all table column
    p_ATTR_RELATION_FACTOR => 311, --http://primes.utm.edu/lists/
    small/1000.txt
    p_CONVERT_TO_ID_FLAG => 'N', --to convert the varchar2 column to id
    column, set this value to 'Y'
    p_COMPARISON_OPERATOR_CN => 'q1026_cp',
    p_LOW_VALUE_CHAR_CN => 'q1026_low_value_char',
    p_HIGH_VALUE_CHAR_CN => NULL,
    p_LOW_VALUE_CHAR_ID_CN => NULL,
    p_LOW_VALUE_NUMBER_CN => NULL,
    p_HIGH_VALUE_NUMBER_CN => NULL,
    p_VALUE1_ID_CN => NULL,
    p_VALUE2_ID_CN => NULL,
    p_VALUE3_ID_CN => NULL,
    p_VALUE4_ID_CN => NULL,
    p_FIRST_CHAR_CN => NULL,
    p_CURRENCY_CODE_CN => NULL,
    p_EQUAL_FLAG => 'Y', -- enable = operator p_BATCH_OP_EQL
    p_BATCH_OP_EQL => '( A.squal_char15 = B.q1026_low_value_char AND
    B.q1026_cp = ''='' )',
    p_LIKE_FLAG => 'N', -- enable LIKE operator p_BATCH_OP_LIKE
    p_BATCH_OP_LIKE => NULL,
    p_BETWEEN_FLAG => 'N', -- enable BETWEEN operator p_BATCH_OP_BETWEEN
    p_BATCH_OP_BETWEEN => NULL,
    p_BATCH_OP_COMMON_WHERE => NULL,
    p_SEEDED_FLAG => NULL,
    p_ALIAS_RULE1 => NULL,
    p_REAL_TIME_SELECT => NULL,
    p_REAL_TIME_WHERE => NULL,
    p_REAL_TIME_FROM => NULL,
    p_HTML_LOV_SQL1 => NULL,
    p_HTML_LOV_SQL2 => NULL,
    p_HTML_LOV_SQL3 => NULL,
    p_HIERARCHY_TYPE => NULL,
    retcode => l_retcode,
    errbuf => l_errbuf);
```

```

update CN_RS_USAGE_DETAILS_ALL
set BATCH_TOTAL_SQL = 'SELECT '||
' T.TRANSACTION_ID TRANS_OBJECT_ID '||
' , T.PARTICIPANT_ID ASSIGNED_OBJECT_ID '||
' , T.SOURCE_EVENT_DATE EVENT_DATE '||
' , T.WORKER_ID '||
' , UPPER(substr(C.ACCOUNT_NAME,1,1)) SQUAL_FC01 '||
' , null SQUAL_CURC01 '||
' , T.SALES_CHANNEL SQUAL_CHAR01 '||
' , T.CITY SQUAL_CHAR02 '||
' , T.STATE SQUAL_CHAR03 '||
' , T.COUNTRY SQUAL_CHAR04 '||
' , C.ACCOUNT_NAME SQUAL_CHAR05 '||
' , P.PARTY_NUMBER SQUAL_CHAR06 '||
' , P.CATEGORY_CODE SQUAL_CHAR07 '||
' , T.POSTAL_CODE SQUAL_CHAR08 '||
' , T.PROVINCE SQUAL_CHAR09 '||
' , T.AREA_CODE SQUAL_CHAR10 '||
' , E.NAME SQUAL_CHAR11 '||
' , T.TRANSACTION_TYPE SQUAL_CHAR12 '||
' , T.TERR_NAME SQUAL_CHAR13 '||
' , R.ROLE_NAME SQUAL_CHAR14 '||
' , T.CUSTOMER_ID SQUAL_NUM01 '||
' , T.INVENTORY_ITEM_ID SQUAL_NUM02 '||
' , T.ELIGIBLE_CAT_ID SQUAL_NUM03 '||
' , T.SOURCE_ORG_ID SQUAL_NUM04 '||
' , T.PARTICIPANT_ID SQUAL_NUM05 '||
' , T.TRANSACTION_AMT_FUNC_CURR SQUAL_NUM06 '||
' , T.TRANSACTION_QTY SQUAL_NUM07 '||
' , T.ORG_ID SQUAL_NUM08 '||
' , null SQUAL_FC02 '||
' , null SQUAL_CURC02 '||
' , T.ATTRIBUTE1 SQUAL_CHAR15 '|| -- IMPORTANT! Update this value to
correspond to index column name
' , NULL SQUAL_CHAR16 '||
' , null SQUAL_CHAR17 '||
' , null SQUAL_CHAR18 '||
' , null SQUAL_CHAR19 '||
' , null SQUAL_CHAR20 '||
' , null SQUAL_NUM09 '||
' , null SQUAL_NUM10 '||
' , null SQUAL_FC03 '||
' , null SQUAL_CURC03 '||
' , null SQUAL_CHAR21 '||
' , null SQUAL_CHAR22 '||
' , null SQUAL_CHAR23 '||
' , null SQUAL_CHAR24 '||
' , null SQUAL_CHAR25 '||
' , null SQUAL_CHAR26 '||
' , null SQUAL_CHAR27 '||
' , null SQUAL_CHAR28 '||
' , null SQUAL_CHAR29 '||
' , null SQUAL_CHAR30 '||
' , null SQUAL_NUM11 '||
' , null SQUAL_NUM12 '||
' , null SQUAL_NUM13 '||
' , null SQUAL_NUM14 '||
' , null SQUAL_NUM15 '||
' , null SQUAL_FC04 '||
' , null SQUAL_CURC04 '||
' , null SQUAL_CHAR31 '||
' , null SQUAL_CHAR32 '||
' , null SQUAL_CHAR33 '||
' , null SQUAL_CHAR34 '||
' , null SQUAL_CHAR35 '||
' , null SQUAL_CHAR36 '||

```

```

' , null SQUAL_CHAR37 '||
' , null SQUAL_CHAR38 '||
' , null SQUAL_CHAR39 '||
' , null SQUAL_CHAR40 '||
' , null SQUAL_NUM16 '||
' , null SQUAL_NUM17 '||
' , null SQUAL_NUM18 '||
' , null SQUAL_NUM19 '||
' , null SQUAL_NUM20 '||
' , null SQUAL_FC05 '||
' , null SQUAL_CURC05 '||
' , null SQUAL_CHAR41 '||
' , null SQUAL_CHAR42 '||
' , null SQUAL_CHAR43 '||
' , null SQUAL_CHAR44 '||
' , null SQUAL_CHAR45 '||
' , null SQUAL_CHAR46 '||
' , null SQUAL_CHAR47 '||
' , null SQUAL_CHAR48 '||
' , null SQUAL_CHAR49 '||
' , null SQUAL_CHAR50 '||
' , null SQUAL_NUM21 '||
' , null SQUAL_NUM22 '||
' , null SQUAL_NUM23 '||
' , null SQUAL_NUM24 '||
' , null SQUAL_NUM25 '||
' from CN_TP_TRANSACTIONS_ALL T ' ||
' , HZ_PARTIES P ' ||
' , HZ_CUST_ACCOUNTS C ' ||
' , JTF_RS_ROLES_VL R ' ||
' , cn_eligible_cats_all_vl E ' ||
' where T.CUSTOMER_ID = C.CUST_ACCOUNT_ID(+) '||
' AND C.PARTY_ID = P.PARTY_ID(+) ' ||
' AND T.ROLE_ID = R.ROLE_ID(+) ' ||
' AND T.ELIGIBLE_CAT_ID = E.ELIGIBLE_CAT_ID(+) '
where usage_id = -1001;

```

```
end;
```

```
commit;
```

```
exit;
```

## Adding an Incentive Compensation Date Qualifier to Crediting Rules

```

declare
  l_retcode VARCHAR2(30);
  l_errbuf VARCHAR2(10000);

begin
  CN_RS_CUST_ATTR.CREATE_QUAL(
    p_attribute_id => -1029, -- unique number in cn_rs_attributes_all table
    p_object_version_number => 1,
    p_org_id => 204, -- operating unit
    p_SOURCE_LANG => 'US',
    p_LANGUAGE => 'US',
    p_name => 'Invoice Date', -- display name
    p_description => 'Custom Attribute: Invoice Date',
    p_ATTR_USAGE_ID => -1029, -- unique number in cn_rs_attr_usages_all
    p_USAGE_ID => -1001,
    p_INDEXED_COL_NAME => 'SQUAL_CHAR19', -- corresponds to
    cn_rs_usage_details_all table batch_total_sql
    p_DISPLAY_TYPE => 'CHAR',
    p_SOURCE_COLUMN_NAME => 'INVOICE_DATE', -- corresponds to
    cn_tp_transactions_all table column
    p_ATTR_RELATION_FACTOR => 331, --http://primes.utm.edu/lists/
    small/1000.txt
    p_CONVERT_TO_ID_FLAG => 'N', --to convert the varchar2 column to id
    column, set this value to 'Y'
    p_COMPARISON_OPERATOR_CN => 'q1029_cp',

```

```

p_LOW_VALUE_CHAR_CN => 'q1029_low_value_char',
p_HIGH_VALUE_CHAR_CN => 'q1029_high_value_char',
p_LOW_VALUE_CHAR_ID_CN => NULL,
p_LOW_VALUE_NUMBER_CN => NULL,
p_HIGH_VALUE_NUMBER_CN => NULL,
p_VALUE1_ID_CN => NULL,
p_VALUE2_ID_CN => NULL,
p_VALUE3_ID_CN => NULL,
p_VALUE4_ID_CN => NULL,
p_FIRST_CHAR_CN => NULL,
p_CURRENCY_CODE_CN => NULL,
p_EQUAL_FLAG => 'Y', -- enable = operator p_BATCH_OP_EQL
p_BATCH_OP_EQL => '( to_date(A.squal_char19,'''YYYY-MM-DD''') =
to_date(B.q1029_low_value_char,'''YYYY-MM-DD''') AND B.q1029_cp =
''='' )',
p_LIKE_FLAG => 'N', -- enable LIKE operator p_BATCH_OP_LIKE
p_BATCH_OP_LIKE => NULL,
p_BETWEEN_FLAG => 'Y', -- enable BETWEEN operator p_BATCH_OP_BETWEEN
p_BATCH_OP_BETWEEN => '( to_date(A.squal_char19,'''YYYY-MM-DD''') BETWEEN
to_date(B.q1029_low_value_char,'''YYYY-MM-DD''') AND
to_date(B.q1029_high_value_char,'''YYYY-MM-DD''') and B.q1029_cp =
''BETWEEN'' )' ,
p_BATCH_OP_COMMON_WHERE => NULL,
p_SEEDED_FLAG => NULL,
p_ALIAS_RULE1 => NULL,
p_REAL_TIME_SELECT => NULL,
p_REAL_TIME_WHERE => NULL,
p_REAL_TIME_FROM => NULL,
p_HTML_LOV_SQL1 => NULL,
p_HTML_LOV_SQL2 => NULL,
p_HTML_LOV_SQL3 => NULL,
p_HIERARCHY_TYPE => NULL,
retcode => l_retcode,
errbuf => l_errbuf);

```

```

update CN_RS_USAGE_DETAILS_ALL
set BATCH_TOTAL_SQL = 'SELECT '''
' T.TRANSACTION_ID TRANS_OBJECT_ID '''
' , T.PARTICIPANT_ID ASSIGNED_OBJECT_ID '''
' , T.SOURCE_EVENT_DATE EVENT_DATE '''
' , T.WORKER_ID '''
' , UPPER(substr(C.ACCOUNT_NAME,1,1)) SQUAL_FC01 '''
' , null SQUAL_CURC01 '''
' , T.SALES_CHANNEL SQUAL_CHAR01 '''
' , T.CITY SQUAL_CHAR02 '''
' , T.STATE SQUAL_CHAR03 '''
' , T.COUNTRY SQUAL_CHAR04 '''
' , C.ACCOUNT_NAME SQUAL_CHAR05 '''
' , P.PARTY_NUMBER SQUAL_CHAR06 '''
' , P.CATEGORY_CODE SQUAL_CHAR07 '''
' , T.POSTAL_CODE SQUAL_CHAR08 '''
' , T.PROVINCE SQUAL_CHAR09 '''
' , T.AREA_CODE SQUAL_CHAR10 '''
' , E.NAME SQUAL_CHAR11 '''
' , T.TRANSACTION_TYPE SQUAL_CHAR12 '''
' , T.TERR_NAME SQUAL_CHAR13 '''
' , R.ROLE_NAME SQUAL_CHAR14 '''
' , T.CUSTOMER_ID SQUAL_NUM01 '''
' , T.INVENTORY_ITEM_ID SQUAL_NUM02 '''
' , T.ELIGIBLE_CAT_ID SQUAL_NUM03 '''
' , T.SOURCE_ORG_ID SQUAL_NUM04 '''
' , T.PARTICIPANT_ID SQUAL_NUM05 '''
' , T.TRANSACTION_AMT_FUNC_CURR SQUAL_NUM06 '''
' , T.TRANSACTION_QTY SQUAL_NUM07 '''
' , T.ORG_ID SQUAL_NUM08 '''
' , null SQUAL_FC02 '''

```

```

' , null SQUAL_CURC02 '||
' , NULL SQUAL_CHAR15 '|| -- IMPORTANT! Update this value to correspond
to index column name
' , NULL SQUAL_CHAR16 '||
' , null SQUAL_CHAR17 '||
' , null SQUAL_CHAR18 '||
' , T.INVOICE_DATE SQUAL_CHAR19 '||
' , null SQUAL_CHAR20 '||
' , null SQUAL_NUM09 '||
' , null SQUAL_NUM10 '||
' , null SQUAL_FC03 '||
' , null SQUAL_CURC03 '||
' , null SQUAL_CHAR21 '||
' , null SQUAL_CHAR22 '||
' , null SQUAL_CHAR23 '||
' , null SQUAL_CHAR24 '||
' , null SQUAL_CHAR25 '||
' , null SQUAL_CHAR26 '||
' , null SQUAL_CHAR27 '||
' , null SQUAL_CHAR28 '||
' , null SQUAL_CHAR29 '||
' , null SQUAL_CHAR30 '||
' , null SQUAL_NUM11 '||
' , null SQUAL_NUM12 '||
' , null SQUAL_NUM13 '||
' , null SQUAL_NUM14 '||
' , null SQUAL_NUM15 '||
' , null SQUAL_FC04 '||
' , null SQUAL_CURC04 '||
' , null SQUAL_CHAR31 '||
' , null SQUAL_CHAR32 '||
' , null SQUAL_CHAR33 '||
' , null SQUAL_CHAR34 '||
' , null SQUAL_CHAR35 '||
' , null SQUAL_CHAR36 '||
' , null SQUAL_CHAR37 '||
' , null SQUAL_CHAR38 '||
' , null SQUAL_CHAR39 '||
' , null SQUAL_CHAR40 '||
' , null SQUAL_NUM16 '||
' , null SQUAL_NUM17 '||
' , null SQUAL_NUM18 '||
' , null SQUAL_NUM19 '||
' , null SQUAL_NUM20 '||
' , null SQUAL_FC05 '||
' , null SQUAL_CURC05 '||
' , null SQUAL_CHAR41 '||
' , null SQUAL_CHAR42 '||
' , null SQUAL_CHAR43 '||
' , null SQUAL_CHAR44 '||
' , null SQUAL_CHAR45 '||
' , null SQUAL_CHAR46 '||
' , null SQUAL_CHAR47 '||
' , null SQUAL_CHAR48 '||
' , null SQUAL_CHAR49 '||
' , null SQUAL_CHAR50 '||
' , null SQUAL_NUM21 '||
' , null SQUAL_NUM22 '||
' , null SQUAL_NUM23 '||
' , null SQUAL_NUM24 '||
' , null SQUAL_NUM25 '||
' from CN_TP_TRANSACTIONS_ALL T ' ||
' , HZ_PARTIES P ' ||
' , HZ_CUST_ACCOUNTS C ' ||
' , JTF_RS_ROLES_VL R ' ||
' , cn_eligible_cats_all_vl E ' ||
' where T.CUSTOMER_ID = C.CUST_ACCOUNT_ID(+) '||

```

```

' AND C.PARTY_ID = P.PARTY_ID(+) ' ||
' AND T.ROLE_ID = R.ROLE_ID(+) ' ||
' AND T.ELIGIBLE_CAT_ID = E.ELIGIBLE_CAT_ID(+) '
where usage_id = -1001;

end;

commit;
exit;

```

## Exposing a Custom Incentive Compensation Attribute in the User Interface: Worked Example

This example demonstrates how to edit and deploy descriptive flexfields so that the application exposes the customized attribute in the user interface (Create Transaction and Manage Transactions pages) and application-generated spreadsheets.

### Editing and Deploying a Descriptive Flexfield

1. If you sign in with the Incentive Compensation Application Administrator duty role, in the global area, select **Administration - Setup and Maintenance**. Continue to step 2.

If you sign in with the Implementation Consultant duty role, click **Manage Descriptive Flexfields** to go to the Manage Descriptive Flexfields page. Skip to step 6.

2. On the Overview page, click **All Tasks**.
3. For **Name**, enter **Manage Incentive Compensation Descriptive Flexfields**.
4. Click **Search**.
5. In the Search Results section, for the returned record, click **Go To Task** to go to the Manage Incentive Compensation Flexfields page.
6. For **Flexfield Code**, enter **CN%**.
7. Click **Search**.
8. In the Search Results section, select the descriptive flexfield record.
9. To add a new attribute, click **Create** to go to the **Create Description Flexfield** page. To edit an existing attribute, click **Edit** to go to the Edit Description Flexfield page.
10. Make your entries or edits.
11. Click **Save and Close** to return to the Manage Incentive Compensation Descriptive Flexfields page.
12. In the Search Results section, click **Deploy Flexfield**.

After you deploy a descriptive flexfield, the new, or edited, attributes appear on all pages that contain the flexfield. It also propagates the attributes to any application-generated spreadsheets that contain the flexfield.

13. Sign out of, and back in to Oracle Fusion Incentive Compensation.

14. Go to either the Create Transaction or Manage Transactions page to see your edits.

## FAQs for Configure Tables and Columns

### How can I enable an attribute to show in the incentive compensation expression builder?

In the Define Business Unit Configuration for Incentive Compensation task list, Configure Tables and Columns task, enable the attribute for calculation and select the appropriate level 2 expression grouping. Also add a user-friendly name for the attribute, that the application displays in the incentive compensation expression builder.

### Why can't I enable an attribute for the incentive compensation Classification or Crediting process?

You must first add the custom qualifiers to the incentive compensation classification or crediting rules.

## Manage Earning Types

### Incentive Compensation Earning Types: Explained

Use earning types to determine the form that participant incentives take. For example, participants associated with plan components where the **Earning Type** is **Monetary Earnings** receive cash compensation for earnings related to these plan components. This is the earning type value that is delivered with the application. It is the only earning type that has monetary value and the only earning type included in incentive compensation payment processing. You can create other earning types and associate them with plan components. The application creates earnings for these plan components and stores them in the database so that they are available for you to use in reporting.

General steps for setting up nonmonetary goals:

1. Create your earning type, for example, **Points**.
2. Create a conversion factor to convert the earning type value to operating currency, for your date range.

---

#### Note

The application uses this conversion factor only for reporting. For example, your operating currency is USD and you define 1 Point = 0.10 operating currency from 1-Jan-10 through 31-Dec-10.

---

#### Restriction

When you create a conversion factor record, if you choose to set an end date, it must be later than the start date.

---



3. Associate your earning type with a plan component, for example you have a plan component Laptop Sales and set the earning type to **Points**.

---

**Tip**

You cannot delete the earning type **Monetary Earnings**. You can delete all other earning types as long as they are not associated with any plan component or used in converting other earning types.

---

When calculation computes the earning, it stores the commission (for example, 500 Points) under COMM\_AMT\_CALC\_CURRENCY in the Earnings entity.

---

**Tip**

Calculation does not do any conversion on nonmonetary earnings (such as **Points**) to participant home currency or operating currency. It stores the nonmonetary balances or earnings, separately in the subledger.

---

Since the payment module does not include nonmonetary earnings, you can export those earnings and set their statuses to **Posted** to ensure that the same nonmonetary earnings are not included the next time that you export them.

When you run the report for the Points earning type, the participant reports show the earnings in the earning type. For example, since the application stored the nonmonetary earnings from our earlier example as 500 Points, along with certain plan component information, the reports show 500 Points.

- Participant reports show only those plan components containing the earning type selected for the report run. Affected reports include: YTD, Earning Details, and Commission Statement.

## Manage Open Period Process

### Incentive Compensation Calendars, Periods, and Period Types: How They Work Together

This topic covers how an incentive compensation calendar supports processing activities for incentive compensation business units by first defining period types and periods, and then exploring how they work together.

#### Period Type

Determines how you divide your calendar or fiscal year. The following table shows the delivered types and corresponding periods in a year:

Period Type	Periods in a Year
Monthly	12
Quarterly	4
Semi-Annually	2
Yearly	1

Create custom period types as required. Enter the number of periods in a year so that the application can validate the periods that you define for the calendar.

Select the shortest required period range (for example, weekly or monthly) to ensure that the application can accumulate attainments, maintain goals and subledger balances, and process payments at this level, or any higher level.

---

**Restriction**

- You cannot accumulate attainments for a period type shorter than your selection. For example, if you select **Monthly**, you cannot accumulate attainments weekly.
- You cannot edit or delete standard or custom period types used by any calendar.

---

**Period**

Define for each calendar based on the selected period type, including the definition of period with year, period name, sequence, and start and end date--inclusive of both the dates. Abbreviate the period name if you want, for example Jan for January or W1 for Week1. The application arranges periods based on the sequence values that you enter.

---

**Restriction**

- The date ranges within one period cannot overlap with other periods and must be continuous with reference to the previous and next periods.
- After you associate a calendar with a business unit and open the first period for the business unit, you cannot edit the calendar.

---

**Tip**

Create all of the periods required for the compensation plans, measures, and goals before creating the plans, measures, and goals.

---

**Calendars and Periods: How They Work**

The following table shows the various calendar and period tasks that first you and then the application perform.

User Actions	Application Actions
1. Create a calendar and associated periods as a single entity within the application.	Store the calendar data in the calendar entity (CN_CALENDARS) and the period data in the period entity (CN_PERIODS).
2. Assign a calendar to a business unit as part of the business unit set up.	<p>Insert the periods (zero or more) that are assigned to that calendar into the business unit level tables. For each interval type defined for the business unit, insert each period into the CN_CAL_PER_INT_TYPES table.</p> <p>The CN_PERIOD_STATUS table stores basic period data, including the default status (Never Opened) given to each period.</p>
3. Edit the period status (for example, select <b>Open</b> ) within the business unit.	Update the period statuses in the CN_PERIOD_STATUS table accordingly. If required, launch a concurrent request to populate the period data in all of the participant-related (SRP) tables.

4. Create new, future periods in the calendar.	<p>Insert the new period records into the following tables:</p> <ul style="list-style-type: none"> <li>• <b>CN_PERIODS:</b> Track the nonbusiness unit based calendar period</li> <li>• <b>CN_PERIOD_STATUS:</b> Track the periods and associated statuses for each business unit. If there are three business units associated with the calendar, insert the newly created period record into all of the business unit tables.</li> <li>• <b>CN_CAL_PER_INT_TYPES:</b> Group each period into the interval type for each business unit. If there are three business units associated with a calendar, and assuming each one has four interval types (<b>Period</b>, <b>Quarter</b>, <b>Semi-Annual</b>, and <b>Year</b>), insert the newly created period record as 12 records (3 business units * 4 interval types).</li> </ul>
--	---

## Incentive Compensation Period Statuses: Explained

There are different statuses that you can edit for incentive compensation periods. The current period status constrains the available values to which you can set the status.

- **Not Opened:** Edit this status to either **Future Entry** or **Open**.

---

### Tip

You cannot edit a period to **Open** if any prior period status is **Not Opened** or **Future Entry**.

- 
- **Future Entry:** Edit this status to **Open**.
  - **Open:** Edit this status to either **Closed** or **Permanently Closed**.

---

### Tip

You cannot edit a period to **Closed** if:

- Any prior period status is **Not Opened**, **Future Entry**, or **Open**
- Any trial payment batches are unpaid (delete or pay the payment batch first, and then close the period)

You cannot edit a period to **Permanently Closed** if any prior period is **Not Opened**, **Future Entry**, **Open**, or **Closed**.

- 
- **Closed:** Edit this status to either **Open** or **Permanently Closed**.
  - **Permanently Closed**

---

### Warning

After you permanently close a period, you cannot reopen it and the application does not process transactions of any kind. Be sure that there are no new transactions, adjustments, payments, or any other outstanding transactions before you permanently close a period.

---

## FAQs for Open Period Process

### Why can't I edit or delete this incentive compensation period?

Most likely, the incentive compensation period was used in one or more of the associated incentive compensation business units, or opened. Or, deleting this period may cause noncontinuous periods, which the application does not allow.

To edit details (such as dates and sequence) for a period that you have not yet opened and did assign to a business unit, delete the period and recreate it in the Define Business Unit Configuration for Incentive Compensation task list, Manage Open Period Process task.

To edit a period that is in the middle of the year, start from the last period defined. For example, if the date range for Sep-09 was wrongly defined and requires correction, starting from Dec-09 (assuming that this is the last defined period), delete the periods Dec-09, Nov-09, and Oct-09 before modifying details for Sep-09.

## Manage Participant Import

### Incentive Compensation Participants: How They Are Imported

Import participants into Oracle Fusion Incentive Compensation using the Staging and Oracle Incentive Compensation Participant Import process, which is located in the Participant Assignments work area, on the Import Participants page.

### Settings That Affect Importing Participants

In the Participant Assignments work area, select the **Run For** choice.

- **Staging and Oracle Incentive Compensation Participant Import:** Use the default integration, which performs the participant import in two phases.
- **Oracle Incentive Compensation Participant Import:** Use your own extraction, transformation, and load utility to populate the staging tables, and then run only the second of the two default phases.

### How Participants Are Imported

There are two phases to the import process, which includes mapping for two Oracle Fusion Incentive Compensation tables.

Phase 1: Import Data to Staging Table (optional)

The first (optional) phase, Staging, runs the default Oracle Data Integrator (ODI) scenario to import the participant data into the CN\_SRP\_PARTICIPANTS\_STAGING\_T staging table. The application only runs this step if you select **Staging and Oracle Incentive Compensation Participant Import** from the **Run For** choice list. It copies all parties from Oracle Fusion Trading Community Architecture to the staging table that meet the specified filter parameters. The application also populates all of the participant attributes,

such as country, currency, and analyst, in the staging table (the mappings are in following sections).

Modify this scenario or create your own script to import data into the staging table, based on your business requirements.

---

**Important**

If you use your own process to import data into the incentive compensation staging table, and use only Phase 2 of the standard import process, then it is possible that the staging table can have invalid data. If columns such as ANALYST\_ID, COUNTRY, CURRENCY\_CODE, or COST\_CENTER are invalid, then you can correct them in the application. There is no way to correct columns such as PARTY\_ID or SOURCE\_SYSTEM\_ID. The application does not expose the column SOURCE\_SYSTEM\_ID and if PARTY\_ID is invalid, then the application will not even display the record.

---

**Phase 2: Import Data to Participant Header Table**

The second phase, Oracle Incentive Compensation Participant Import, collects the data into the participant header table CN\_SRP\_PARTICIPANTS\_ALL and participant detail table CN\_SRP\_PARTICIPANT\_DETAILS\_ALL. Next, for the imported parties, if there is no party usage code specified, it inserts a record into the HZ\_PARTY\_USG\_ASSIGNMENTS table with a party usage code of INCENTIVE\_COMP\_PARTICIPANT. Last, the application deletes the party records that were imported in phase 1, from the CN\_SRP\_PARTICIPANTS\_STAGING\_T staging table.

---

**Restriction**

The application will not copy any record where the combination of PART\_ID and ORG\_ID already exists in the CN\_SRP\_PARTICIPANTS\_ALL header table.

---

**Mapping for CN\_SRP\_PARTICIPANTS\_ALL**

This table maps each CN\_SRP\_PARTICIPANTS\_ALL column name to the corresponding Oracle Fusion Trading Community or Oracle Fusion Human Capital Management (HCM) column name and includes comments about the values.

Column name in CN_SRP_PARTICIPANTS_ALL	Column Name in Oracle Fusion Trading Community or HCM	Comments
PARTICIPANT_ID	None	Oracle Data Integrator generates it from the sequence CN_SRP_PARTICIPANTS_S1.
PARTY_ID	HZ_PARTIES.PARTY_ID	
START_DATE	PER_ALL_PEOPLE_F.START_DATE HZ_PARTY_USG_ASSIGNMENTS	Oracle Data Integrator first tries to use the target parameter Active Start Date first. If it is null, then Oracle Data Integrator uses either PER_ALL_PEOPLE_F.START_DATE or HZ_PARTY_USG_ASSIGNMENTS.EFFECTIVE_START_DATE
END_DATE	None	Oracle Data Integrator does not populate it.
ACTIVE_FLAG	None	Oracle Data Integrator always populated with Y (yes).

ANALYST_ID	None	Oracle Data Integrator first tries to use the target parameter Analyst. If it is NULL, then ODI does not populate ANALYST_ID.
PARTICIPANT_TYPE	None	Oracle Data Integrator uses the target parameter Participant Type, which defaults to Participant.
COMPENSATION_END_DATE	None	Oracle Data Integrator does not populate it.
HOLD_PAYMENT_FLAG	None	Oracle Data Integrator always populated with N (no).
HOLD_REASON	None	Oracle Data Integrator does not populate it.
DISPLAY_IDENTIFIER	None	Oracle Data Integrator does not populate it.
SOURCE_SYSTEM	None	Not applicable.
SOURCE_SYSTEM_ID	PER_ALL_PEOPLE_F.PERSON_ID	If there is no entry in Oracle Fusion HCM, then Oracle Data Integrator populates it with NULL.
ORG_ID	None	Oracle Data Integrator uses the target parameter Business Unit.
PARTY_NUMBER	HZ_PARTIES.PARTY_NUMBER	Oracle Data Integrator populates it.
HR_PRIMARY_WORKER_NUMBER	HZ_PARTIES.HR_PRIMARY_WORKER_NUMBER	Oracle Data Integrator populates it.
PAYEE_ONLY	None	Oracle Data Integrator does not populate it.
USER_GUID	None	Oracle Data Integrator does not populate it.

#### Mapping for CN\_SRP\_PARTICIPANT\_DETAILS\_ALL

This table maps each CN\_SRP\_PARTICIPANT\_DETAILS\_ALL to the corresponding Oracle Fusion Trading Community column name and includes comments about the values.

CN_SRP_PARTICIPANT_DETAIL	Column Name in Oracle Fusion Trading Community	Comments
PARTICIPANT_DETAIL_ID	None	Oracle Data Integrator generates it from the sequence CN_SRP_PARTICIPANTS_S1.
PARTICIPANT_ID	None	Oracle Data Integrator populates it with CN_SRP_PARTICIPANTS_ALL.PARTICIPANT_ID.
START_DATE	None	Oracle Data Integrator populates it with CN_SRP_PARTICIPANTS_ALL.START_DATE.
END_DATE	None	Oracle Data Integrator populates it with CN_SRP_PARTICIPANTS_ALL.END_DATE.

ORG_ID	None	Oracle Data Integrator populates it with CN_SRP_PARTICIPANTS_ALL.ORG_ID.
COST_CENTER	None	Oracle Data Integrator does not populate it.
COUNTRY_CODE	HZ_PARTIES.COUNTRY_CODE	If the Oracle Fusion Trading Community column is NULL, then Oracle Data Integrator tries to use the target parameter Target Country. If it is also NULL, then Oracle Data Integrator does not populate COUNTRY_CODE.
CURRENCY_CODE	None	Oracle Data Integrator first tries to use the target parameter Target Participant Home Currency. If it is NULL, then Oracle Data Integrator populates it with the operating currency of the ORG_ID (CN_REPOSITORIES_ALL_B.FUNCTIONAL_CURRENCY).

## Incentive Compensation Import Participants Parameters

One of the steps during implementation is to establish the people who receive incentives. In the Participant Assignment work area, on the Import Participants page the application enables you to search person, party, and resource information in the Oracle Fusion Human Capital Management (HCM) Foundation, Oracle Fusion Trading Community, or Oracle Fusion Resource Manager common objects, then select one or more individuals to import.

There are two types of parameters for you to use when importing participants:

- **Target:** During the Staging and Oracle Incentive Compensation Participant Import process, when the application collects data into the staging table, it runs a script that attempts to map all participant attributes from the source and insert those values (for example, Active Start Date and Currency) into the staged record. When step two of the import process runs, the application applies any selected target parameters to the data it imports into the incentive compensation participant tables, when the value for that attribute is null.
- **Filter:** The import process uses the parameters provided in the filter region as part of a query to run for the selection process. Select one or more of the available parameters to filter selection criteria.

### *Target Parameters*

#### **Business Unit**

Required. Set the incentive compensation business unit into which the application imports the participants. During step one of the process, Oracle Data Integrator leaves the staging table value as NULL and for step two, uses it to populate CN\_SRP\_PARTICIPANTS\_ALL.ORG\_ID and CN\_SRP\_PARTICIPANT\_DETAILS\_ALL.ORG\_ID.

#### **Currency**

Set it as a default currency during the import if the source application (and the incentive compensation staging table) does not contain this value. You can reset this parameter for each import.

- If this target parameter has a value, Oracle Data Integrator uses it to populate CN\_SRP\_PARTICIPANT\_DETAILS\_ALL.CURRENCY\_CODE.
- If the value is NULL in staging, and you do not set the target parameter, Oracle Data Integrator uses the operating currency from CN\_REPOSITORIES\_ALL\_B.

#### **Target Participant Home Currency**

- If this target parameter has a value, Oracle Data Integrator uses it to populate CN\_SRP\_PARTICIPANT\_DETAILS\_ALL.CURRENCY\_CODE.
- If the staged record currency value is NULL, and you did not select a target parameter, Oracle Data Integrator uses the operating currency from CN\_REPOSITORIES\_ALL\_B.

#### **Analyst Name**

- If this target parameter has a value, Oracle Data Integrator uses it to populate CN\_SRP\_PARTICIPANTS\_ALL.ANALYST\_ID.
- If there is no value, Oracle Data Integrator populates CN\_SRP\_PARTICIPANTS\_ALL.ANALYST\_ID with NULL.

#### **Target Country**

Get the value from the person record in Oracle Fusion Human Capital Management (HCM) or party location for primary address.

- If the application does not find one, it leaves the value null. Oracle Data Integrator first tries to use the Oracle Fusion Trading Community Architecture value HZ\_PARTIES.COUNTRY to populate CN\_SRP\_PARTICIPANT\_DETAILS\_ALL.COUNTRY\_CODE.
- If it is NULL, then Oracle Data Integrator uses this target parameter value. If the target parameter is also NULL, then CN\_SRP\_PARTICIPANT\_DETAILS\_ALL.COUNTRY\_CODE is NULL.

#### **Active Start Date**

Get the start date from either the PER\_ALL\_PEOPLE\_F.START\_DATE or HZ\_PARTIES.CREATION\_DATE column. Oracle Data Integrator uses it to populate CN\_SRP\_PARTICIPANTS\_ALL.START\_DATE and CN\_SRP\_PARTICIPANT\_DETAILS\_ALL.START\_DATE

#### **Participant Type**

Defaults to PARTICIPANT.

#### ***Filter Parameters***

---

##### **Important**

Use filter parameters whenever possible, to avoid importing extraneous participants because there is no way to delete them after import.

---



## Party Usage

Required. Defaults to Incentive Compensation, if you select a value, then Oracle Data Integrator matches it against the value in HZ\_PARTY\_USG\_ASSIGNMENTS.PARTY\_USAGE\_CODE, where the HZ\_PARTY\_USG\_ASSIGNMENTS.STATUS\_FLAG value is A (active). Oracle Data Integrator uses the LIKE operator to support wildcard characters.

## Party Name

Text input, with wildcard support. If you provide a value, Oracle Data Integrator applies the filter condition: UPPER(HZ\_PARTIES.PARTY\_NAME) LIKE UPPER(parameter value).

## HCM Job

Text input, with wildcard support. Oracle Data Integrator uses the following SQL to filter the parties based on job code.

```
Select *
From hz_parties hp, hz_orig_sys_references ref, per_all_assignments_m
  assgn, per_jobs_f job
Where ref.Orig_System = 'FUSION_HCM'
And ref.Owner_Table_Name = 'HZ_PARTIES'
And hr.Owner_Table_Id = hp.Party_Id
And hp.Orig_System_Reference = assgn.Person_Id
And Sysdate between assgn.Effective_Start_Date and
  assgn.Effective_End_Date
And assgn.Job_Id = job.Job_Id
And job.Job_Code Like Upper('parameter value');
```

## Country

Text Input, without wildcard support. If you provide a value, Oracle Data Integrator matches it against the value in HZ\_PARTIES.COUNTRY.

## Start Date

Required. Oracle Data Integrator matches it against the value in HZ\_PARTY\_USG\_ASSIGNMENTS.EFFECTIVE\_START\_DATE.

## End Date

Oracle Data Integrator matches it against the value in HZ\_PARTY\_USG\_ASSIGNMENTS.EFFECTIVE\_END\_DATE.

## Role

Text input, with wildcard support. If you provide a value, Oracle Data Integrator matches it against the Oracle Fusion Resource Manager value JTF\_RS\_ROLES\_B.ROLE\_CODE.

## Role Type

Text input, without wildcard support. If you provide a value, Oracle Data Integrator matches it against the value JTF\_RS\_ROLES\_B.ROLE\_TYPE.

## Person Number

Text input, without wildcard support. If you provide a value, Oracle Data Integrator matches it against the Oracle Fusion HCM value PER\_ALL\_PEOPLE\_F.PERSON\_NUMBER.



---

## Define Transactional Business Intelligence Configuration

### 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.

Product	Job or Abstract Role	Security Profile
Oracle Fusion Assets	Asset Accountant	View All Workers

Oracle Fusion Assets	Asset Accounting Manager	View All Workers
Oracle Fusion Enterprise Planning and Budgeting	Budget Analyst	View All Workers
Oracle Fusion Enterprise Planning and Budgeting	Budget Manager	View All Workers
Oracle Fusion Financial Consolidation Hub	Consolidation Accountant	View All Workers
Oracle Fusion Financial Consolidation Hub	Consolidation Manager	View All Workers
Oracle Fusion Financials Common Module	Intercompany Accountant	View All Workers
Oracle Fusion General Ledger	Financial Analyst	View All Workers
Oracle Fusion General Ledger	General Accountant	View All Workers
Oracle Fusion General Ledger	General Accounting Manager	View All Workers
Oracle Fusion Incentive Compensation	Incentive Compensation Participant Manager	View Manager Hierarchy
Oracle Fusion Inventory Management	Warehouse Manager	View All Workers
Oracle Fusion Project Foundation	Project Accountant	View All Workers
Oracle Fusion Project Foundation	Project Administrator	View All Workers
Oracle Fusion Project Foundation	Project Billing Specialist	View All Workers
Oracle Fusion Purchasing	Buyer	View All Workers
Oracle Fusion Sourcing	Category Manager	View All Workers
Oracle Fusion Sourcing	Sourcing Project Collaborator	View All Workers

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.

Decisions to Consider	In This Example
What is the name of the Transactional Business Intelligence job or abstract role?	Warehouse Manager
What is the name of the person security profile?	View All Workers

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:

Field	Value
<b>Search</b>	Tasks
<b>Name</b>	Manage Data Role and Security Profiles

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**.



---

## Define Extensions and Page Customizations

### Define Custom Enterprise Scheduler Jobs

#### 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

## Customization and Sandboxes

### 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.
- 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

#### Global Page Template

- Customize the global area at the site layer.

See: Editing the UI Shell Template Used by All Pages

#### 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

## Sandboxes

- 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.

Type of Changes	Type of Sandbox	Method for Making Changes Available in Mainline	Downloadable?
Metadata	Sandbox	Publish sandbox	Yes
Data security	Sandbox enabled for data security changes	Publish sandbox	Yes
Flexfield	Flexfield deployed as a flexfield-enabled sandbox	Deploy flexfield	No

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

## **FAQs for Page Customizations**

### **How can I change the web mapping service for displaying contextual addresses?**

Edit the Mapping Service for Contextual Addresses profile option value. A contextual address is marked with an orange square icon that can be clicked to display the address on a map. The profile option value represents the web mapping service used to display the map. To update this value, use the Manage Administrator Profile Values task in the Setup and Maintenance work area.

---

# 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.

---

**Note**

Only the translation in the current user language is moved.

---

Business Object Name	Moved Functional Item	Effect on the Scope of Movement
Application Menu Customization	Customizations to the navigator menu	Movement of navigator menu customizations requires manual effort. For details regarding the manual command, refer to the Oracle Fusion Applications Administrators Guide.
Application Message	Messages and associated tokens	No parameters: all messages are moved.  moduleType/moduleKey: only messages belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.  messageName/applicationId: only the specified message is moved.
Application Taxonomy	Application taxonomy modules and components	No parameters: all taxonomy modules and components are moved.
Application Attachment Entity	Attachment entities	No parameters: all attachment entities are moved.  moduleType/moduleKey: only attachment entities belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.
Application Attachment Category	Attachment categories and category-to-entity mappings	No parameters: all attachment categories and category-to-entity mappings are moved.  moduleType/moduleKey: 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.
Application Document Sequence Category	Document sequence categories	No parameters: all categories are moved.  moduleType/moduleKey: only categories belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.  code/applicationId: only the specified document sequence category code is moved.

Application Document Sequence	Document sequences and their assignments	<p>No parameters: all sequences are moved.</p> <p><code>moduleType/moduleKey</code>: only document sequences belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved</p> <p><code>name</code>: only the specified document sequence is moved.</p>
Application Descriptive Flexfield	Descriptive flexfield registration data and setup data	<p>No parameters: all descriptive flexfields are moved.</p> <p><code>moduleType/moduleKey</code>: only descriptive flexfields belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>descriptiveFlexfieldCode/applicationId</code>: only the specified descriptive flexfield is moved.</p> <hr/> <p><b>Note</b></p> <p>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.</p> <hr/> <p><b>Note</b></p> <p>Only flexfields with a deployment status of Deployed or Deployed to Sandbox are eligible to be moved.</p> <hr/>



Application Extensible Flexfield	Extensible flexfield registration data and setup data, including categories	<p>No parameters: all extensible flexfields are moved</p> <p><code>moduleType/moduleKey</code>: only extensible flexfields belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>extensibleFlexfieldCode/applicationId</code>: only the specified extensible flexfield is moved.</p> <hr/> <p><b>Note</b></p> <p>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.</p> <hr/> <p><b>Note</b></p> <p>Only flexfields with a deployment status of Deployed or Deployed to Sandbox are eligible to be moved.</p> <hr/>
Application Key Flexfield	Key flexfield registration data and setup data	<p>No parameters: all key flexfields are moved.</p> <p><code>moduleType/moduleKey</code>: only key flexfields belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>keyFlexfieldCode/applicationId</code>: only the specified key flexfield is moved.</p> <hr/> <p><b>Note</b></p> <p>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.</p> <hr/> <p><b>Note</b></p> <p>Only flexfields with a deployment status of Deployed or Deployed to Sandbox are eligible to be moved.</p> <hr/>

Application Flexfield Value Set	Value set setup data	<p>No parameters: all value sets are moved.</p> <p><code>moduleType/moduleKey</code>: only value sets belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>valueSetCode</code>: only the specified value set is moved.</p> <hr/> <p><b>Note</b></p> <p>Importing a value set's metadata can change the deployment status of flexfields that use the value set, and therefore the affected flexfields must be redeployed. The import process automatically submits affected flexfields for redeployment.</p> <hr/>
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	<p>No parameters: all standard lookups are moved.</p> <p><code>moduleType/moduleKey</code>: only standard lookups belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>lookupType</code>: only the specified common lookup is moved.</p>

Application Common Lookup	Common lookup types and their lookup codes	<p>No parameters: all common lookups are moved.</p> <p><code>moduleType/moduleKey</code> - only common lookups belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>lookupType</code>: only the specified common lookup is moved.</p>
Application Set-Enabled Lookup	Set-enabled lookup types and their lookup codes	<p>No parameters: all set-enabled lookups are moved.</p> <p><code>moduleType/moduleKey</code>: only set-enabled lookups belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>lookupType</code>: only the specified set-enabled lookup is moved.</p>
Application Profile Category	Profile categories	<p>No parameters: all profile categories are moved.</p> <p><code>moduleType/moduleKey</code>: only categories belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>name/applicationId</code>: only the specified category is moved.</p>
Application Profile Option	Profile options and their values	<p>No parameters: all profile options and their values are moved.</p> <p><code>moduleType/moduleKey</code>: only profile options and their values belonging to the specified module are moved.</p> <p><code>profileOptionName</code>: only the specified profile option and its values are moved.</p>

Application Profile Value	Profile options and their values	<p>No parameters: all profiles and their values are moved.</p> <p><code>moduleType/moduleKey</code>: only profiles and their values belonging to the specified module are moved.</p> <p><code>categoryName/categoryApplicationId</code>: only profiles and their values belonging to the specified category are moved.</p> <p><code>profileOptionName</code>: only the specified profile and its values are moved.</p>
Application Reference Data Set	Reference data sets	No parameters: all sets are moved.
Application Reference Data Set Assignment	Reference data set assignments	<p><code>determinantType</code>: only assignments for the specified determinant type are moved.</p> <p><code>determinantType/referenceGroupName</code>: only assignments for the specified determinant type and reference group are moved.</p>
Application Tree Structure	Tree structures and any labels assigned to the tree structure	<p>No parameters: all tree structures (and their labels) are moved.</p> <p><code>moduleType/moduleKey</code>: only tree structures (and their labels) belonging to the specified module are moved.</p> <p><code>treeStructureCode</code>: only the specified tree structure (with its labels) is moved.</p>
Application Tree	Tree codes and versions	<p>No parameters: all trees are moved.</p> <p><code>moduleType/moduleKey</code>: only trees belonging to the specified module are moved.</p> <p><code>treeStructureCode</code>: only trees belonging to the specified tree structure are moved.</p> <p><code>TreeStructureCode/TreeCode</code>: only trees belonging to the specified tree structure and tree code are moved.</p>

Application Tree Label	Tree structures and any labels assigned to the tree structure	<p>No parameters: all tree structures (and their labels) are moved.</p> <p><code>moduleType/moduleKey</code>: only tree structures (and their labels) belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>treeStructureCode</code>: only the specified tree structure (with its labels) is moved.</p>
Application Data Security Policy	Database resources, actions, conditions, and data security policies	<p>No parameters: all database resources/actions/conditions/policies are moved.</p> <p><code>moduleType/moduleKey</code>: only database resources/actions/conditions/policies belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>objName</code>: only the specified database resource along with its actions/conditions/policies is moved.</p> <hr/> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>• If the policies being moved contain reference to newly created roles, move the roles before moving the policies.</li> <li>• If the source and target systems use different LDAPs, manually perform the GUID reconciliation after moving the data security policies.</li> </ul>
Application Activity Stream Configuration	Activity stream options	No parameters: all activity stream options are moved.

## 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.





---

# Glossary

## **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.

## **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.

## **action**

The kind of access named in a security policy, such as view or edit.

## **ADF**

Acronym for Application Developer Framework. A set of programming principles and rules for developing software applications.

## **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.

## **attainment**

Indicates the participant's achievement against a specified target, for a specified performance measure. Typically, add qualified credit transactions to get an attainment total expressed in any unit of measure, such as amount or percent. For example, attained sales of 200,000 USD this month on a target of 250,000 USD.

**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**

Defines the incentive compensation processing periods by defining the calendar, period type, and period. It does not contain an end date. The plan administrator can add more periods and must associate the calendar with a business unit before the application can use the periods in processing for the business unit.

**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.

**collect transactions**

Move transaction data from the staging table to the transaction (CN\_CP\_TRANSACTIONS\_ALL) table after checking for duplicates, validating data, and converting currencies.

**condition**

An XML filter or SQL predicate WHERE clause in a data security policy that specifies what portions of a database resource are secured.

**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.

**cost center**

A unit of activity or group of employees used to assign costs for accounting purposes.

**cost organization**

A grouping of inventory organizations that indicates legal and financial ownership of inventory, and which establishes common costing and accounting policies.

**country holding company**

A legal entity that acts on behalf of several divisions within an enterprise, and is the legal employer in a country.

**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.

**department**

A division of a business enterprise dealing with a particular area of activity.

**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.

**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.

**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.

**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.

**goal**

A performance objective, also known as a quota. The best practice is to make it specific, measurable, achievable, and time based (SMART). Define your goal (optional), by providing a target number and the unit of measure (either amount or quantity).

**grade**

A component of the employment model that defines the level of compensation for a worker.

**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.

**incentive compensation business unit**

A central incentive compensation administration center, which administers incentive compensation for a group of participants. Associate participants within a single business unit with different business organizations or countries.

**interface table**

A database table used for transferring data between applications or from an external application or data file.

**inventory organization**

An organization that tracks inventory transactions and balances, and can manufacture or distribute products.

**item master**

A collection of data that describes items and their attributes recorded in a database file.

**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.

**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.

**model profile**

A collection of the work requirements and required skills and qualifications of a workforce structure, such as a job or position.

**natural account**

Categorizes account segment values by account type, asset, liability, expense, revenue, or equity, and sets posting, budgeting, and other options.

**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.

**operating currency**

Defines which currency to use for the incentive compensation business unit.

**participant**

A person or organization (for example, an employee, salesperson, party, supplier contract, partner, or third party resale contractor) whose credits, attainment, earnings, disputes, and payments the application computes and manages.



**participant home currency**

Defines which currency to use for each participant.

**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.

**period type**

Is the shortest period range that must be available for incentive compensation processing, for example monthly. Associate the period type with a calendar and define incentive compensation periods (for example, Jan-10, Feb-10) for a calendar based on it.

**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.

**processing currency**

Determines whether the application uses either operating currency or participant home currency when processing credits, earnings, and payments.

**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.

**project expenditure organization**

An organization that can incur expenditures and hold financial plans for projects.

**rate**

The output of a rate table (an amount or a percentage) that the application often applies against the credit amount, or target incentive, to determine the amount to pay the participant. For example, pay 5 percent (the rate) of all sales revenue to Salesperson A on a monthly basis.

**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.

**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.

**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.

**SOA**

Abbreviation for service-oriented architecture.

**SQL predicate**

A type of condition using SQL to constrain the data secured by a data security policy.

**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.

**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.

**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.