# Contents

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

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

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

## 4 Define Implementation Users
- Initial Security Administration: Critical Choices ....................................................... 4-1
- Initial Security Administration: Worked Example ..................................................... 4-3

## 5 Define Currencies and Currency Rates
- Manage Currencies ....................................................................................................... 5-1
- Manage Conversion Rate Types .................................................................................. 5-1
- FAQs for Manage Conversion Rate Types ................................................................... 5-4

## 6 Define Enterprise Structures
- Enterprise Structures: Overview ................................................................................ 6-1
- Enterprise Structures Business Process Model: Explained ....................................... 6-3
- Global Enterprise Configuration: Points to Consider ............................................... 6-5
- Modeling Your Enterprise Management Structure in Oracle Fusion: Example ........ 6-6
- Essbase Character and Word Limitations ................................................................... 6-9
- Define Initial Configuration with the Enterprise Structures Configurator .................. 6-11
- Define Reference Data Sharing .................................................................................. 6-36
- Define Enterprise: Manage Enterprise HCM Information ....................................... 6-43
- Define Enterprise: Manage Locations .......................................................................... 6-44
- Define Geographies ...................................................................................................... 6-46
7 Define Social Networking

How can I enable social networking features? .......................................................... 7-1

8 Define Security for Supply Chain Management

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 .................. 8-7
Security Tasks: Overview ....................................................................................... 8-10
Define Data Security .............................................................................................. 8-14
Manage Role Templates .......................................................................................... 8-18
Define Users ......................................................................................................... 8-20
Import Worker Users ............................................................................................. 8-23
Manage Users ......................................................................................................... 8-27
FAQs for Manage Users .......................................................................................... 8-29

9 Define Automated Governance, Risk, and Performance Controls

Segregation of Duties: Explained ............................................................................ 9-1
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defining Segregation of Duties Policies: Points To Consider</td>
<td>9-4</td>
</tr>
<tr>
<td>Managing Segregation of Duties Risks and Violations: Critical Choices</td>
<td>9-6</td>
</tr>
<tr>
<td>Role Provisioning and Segregation of Duties: How They Work Together</td>
<td>9-8</td>
</tr>
<tr>
<td>10 Define Approval Management for Supply Chain Management</td>
<td>10-1</td>
</tr>
<tr>
<td>Approval Management: Highlights</td>
<td></td>
</tr>
<tr>
<td>11 Define Help Configuration</td>
<td>11-1</td>
</tr>
<tr>
<td>Define Help Configuration: Overview</td>
<td></td>
</tr>
<tr>
<td>Set Help Options</td>
<td></td>
</tr>
<tr>
<td>FAQs for Assign Help Text Administration Duty</td>
<td></td>
</tr>
<tr>
<td>Manage Help Security Groups</td>
<td></td>
</tr>
<tr>
<td>12 Define Source Systems</td>
<td>12-1</td>
</tr>
<tr>
<td>Source Systems: Explained</td>
<td></td>
</tr>
<tr>
<td>Source System Entities: Explained</td>
<td></td>
</tr>
<tr>
<td>FAQs for Define Source Systems</td>
<td></td>
</tr>
<tr>
<td>13 Define Application Toolkit Configuration</td>
<td>13-1</td>
</tr>
<tr>
<td>Define Application Toolkit Configuration: Overview</td>
<td></td>
</tr>
<tr>
<td>Map Reports to Work Areas</td>
<td></td>
</tr>
<tr>
<td>Set Watchlist Options</td>
<td></td>
</tr>
<tr>
<td>14 Common Applications Configuration: Maintain Common Reference Objects</td>
<td>14-1</td>
</tr>
<tr>
<td>Maintain Common Reference Objects: Overview</td>
<td></td>
</tr>
<tr>
<td>Define Application Taxonomy</td>
<td></td>
</tr>
<tr>
<td>Define Reference Data Sharing</td>
<td></td>
</tr>
<tr>
<td>Define ISO Reference Data</td>
<td></td>
</tr>
<tr>
<td>Manage Data Security Policies</td>
<td></td>
</tr>
<tr>
<td>Set Activity Stream Options</td>
<td></td>
</tr>
<tr>
<td>Manage Menu Customizations</td>
<td></td>
</tr>
<tr>
<td>Manage Audit Policies</td>
<td></td>
</tr>
<tr>
<td>Manage Oracle Social Network Objects</td>
<td></td>
</tr>
<tr>
<td>Manage Applications Core Common Reference Objects: Manage Applications Core Messages</td>
<td></td>
</tr>
<tr>
<td>Manage Applications Core Common Reference Objects: Manage Applications Core Administrator Profile Values</td>
<td></td>
</tr>
<tr>
<td>FAQs for Manage Applications Core Administrator Profile Values</td>
<td></td>
</tr>
<tr>
<td>How can I enable the privacy statement?</td>
<td></td>
</tr>
<tr>
<td>15 Define WebLogic Communication Services Configuration</td>
<td></td>
</tr>
</tbody>
</table>
### 16 Define Applications Core Configuration

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define Lookups</td>
<td>16-1</td>
</tr>
<tr>
<td>Manage Messages</td>
<td>16-8</td>
</tr>
<tr>
<td>Define Document Sequences</td>
<td>16-10</td>
</tr>
<tr>
<td>Define Trees</td>
<td>16-13</td>
</tr>
<tr>
<td>Define Profile Options</td>
<td>16-29</td>
</tr>
<tr>
<td>Define Flexfields</td>
<td>16-38</td>
</tr>
<tr>
<td>Define Attachments</td>
<td>16-122</td>
</tr>
</tbody>
</table>

### 17 Define Product Information Management Configuration

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Information Management Features by License: Overview</td>
<td>17-1</td>
</tr>
<tr>
<td>Define Basic Items</td>
<td>17-2</td>
</tr>
<tr>
<td>Product Child Value Sets: Explained</td>
<td>17-2</td>
</tr>
<tr>
<td>Managing Default Item Class: Explained</td>
<td>17-3</td>
</tr>
<tr>
<td>Item Statuses: Explained</td>
<td>17-3</td>
</tr>
<tr>
<td>Managing Item Types: Explained</td>
<td>17-4</td>
</tr>
<tr>
<td>Cross-Reference Types: Explained</td>
<td>17-4</td>
</tr>
<tr>
<td>Item Imports: Explained</td>
<td>17-4</td>
</tr>
<tr>
<td>Related Item Subtypes: Explained</td>
<td>17-5</td>
</tr>
<tr>
<td>Managing Descriptive Flexfields for Items: Explained</td>
<td>17-5</td>
</tr>
<tr>
<td>FAQs for Define Basic Items</td>
<td>17-7</td>
</tr>
<tr>
<td>What's the difference between lifecycle phase types and lifecycle phases?</td>
<td>17-7</td>
</tr>
<tr>
<td>Define Basic Product Structure</td>
<td>17-8</td>
</tr>
<tr>
<td>Item Structure Type Details: Explained</td>
<td>17-8</td>
</tr>
<tr>
<td>Define Basic Catalogs</td>
<td>17-9</td>
</tr>
<tr>
<td>Items to Categories Assignment: Explained</td>
<td>17-10</td>
</tr>
<tr>
<td>Catalog Publishing: Explained</td>
<td>17-11</td>
</tr>
<tr>
<td>FAQs for Define Basic Catalogs</td>
<td>17-11</td>
</tr>
<tr>
<td>Define Advanced Items</td>
<td>17-12</td>
</tr>
<tr>
<td>Transactional Attributes: Explained</td>
<td>17-13</td>
</tr>
<tr>
<td>FAQs for Define Advanced Items</td>
<td>17-16</td>
</tr>
<tr>
<td>Define Change Orders</td>
<td>17-16</td>
</tr>
<tr>
<td>Define Product Rules</td>
<td>17-17</td>
</tr>
<tr>
<td>What's rule set assigning?</td>
<td>17-18</td>
</tr>
<tr>
<td>What's a return type for?</td>
<td>17-19</td>
</tr>
<tr>
<td>What's a rule sequence?</td>
<td>17-19</td>
</tr>
<tr>
<td>What's a rule-builder expression?</td>
<td>17-19</td>
</tr>
<tr>
<td>What's a severity equals warning?</td>
<td>17-19</td>
</tr>
<tr>
<td>What's a validation condition?</td>
<td>17-19</td>
</tr>
<tr>
<td>What's the difference between assignment rule sets and validation rule sets?</td>
<td>17-20</td>
</tr>
<tr>
<td>What's the hierarchy of rules and rule sets?</td>
<td>17-20</td>
</tr>
<tr>
<td>What's the user message and where does it display?</td>
<td>17-20</td>
</tr>
<tr>
<td>Define New Item Requests</td>
<td>17-20</td>
</tr>
</tbody>
</table>
21 Importing and Exporting Setup Data

Configuration Packages: Explained ................................................................. 21-1
Exporting and Importing Setup Data: Explained ............................................. 21-1
Moving Common Reference Objects .............................................................. 21-2
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.

![Help Icon](image)

**Note**

If you don’t see any help icons on your page, then click the Show Help icon button in the global area. However, not all pages have help icons.

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. To access the guides, go to any page in Oracle Fusion Applications Help and select **Documentation Library** from the **Navigator** menu.

Guides are designed for specific audiences:

- **User Guides** address the tasks in one or more business processes. They are intended for users who perform these tasks, and managers looking for an overview of the business processes. They are organized by the business process activities and tasks.

- **Implementation Guides** address the tasks required to set up an offering, or selected features of an offering. They are intended for implementors. They are organized to follow the task list sequence of the offerings, as displayed within the Setup and Maintenance work area provided by Oracle Fusion Functional Setup Manager.

- **Concept Guides** explain the key concepts and decisions for a specific area of functionality. They are intended for decision makers, such as chief
financial officers, financial analysts, and implementation consultants. They are organized by the logical flow of features and functions.

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

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

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

**Note**
Limited content applicable to Oracle Cloud implementations.

For other guides, 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 details on 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 Oracle Enterprise Repository 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.

- Other technical information such as reusable components, policies, architecture diagrams, and topology diagrams.

Documentation Accessibility

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

Comments and Suggestions

Your comments are important to us. We encourage you to send us feedback about Oracle Fusion Applications Help and guides. Please send your suggestions to oracle_fusion_applications_help_ww_grp@oracle.com. You can use Send Feedback to Oracle from the Settings and Actions menu in Oracle Fusion Applications Help.
Overview

Product Management Offering: Overview

By implementing the Product Management offering, your enterprise can configure how to create and manage items, catalogs, and related processes, as well as the structures to support product innovation management. The Oracle Fusion Applications Product Information Management Implementation Guide covers implementation of item, catalog, and product management processes, while the Oracle Fusion Innovation Management Implementation Guide covers implementation of processes related to ideas and requirements management, concept design management, and product portfolio management.

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

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

- Product Management (this option also has associated features that you can select to configure)
- Product and Catalog Management
- Basic Product Structures
- Basic Catalogs
- Change Orders
- Product Rules
- Advanced Product Structures
- New Item Requests
- Product Hub
• Product Source Systems
• Advanced Catalogs
• Data Quality for Products
• Product Information Management Business Intelligence Analytics
• Product Management Business Intelligence Analytics
• Product Requirements and Ideation Management
• Concept Design Management
• Product Lifecycle Portfolio Management

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

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

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

**Define Common Applications Configuration for Product Management**

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

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

**Define Product Information Management Configuration**

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

**Define Product Innovation**

Use this task list to configure product innovation objects such as ideas, requirements, feature sets, concepts, proposals, and portfolios, and to set up integration with target systems.

**Define Transactional Business Intelligence Configuration**

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

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

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

Getting Started with an Implementation: Overview

To start an Oracle Fusion Applications implementation, you must set up one or more initial users using the super user that was created during installation and provisioning of the Oracle Fusion Applications environment, or using the initial administrator user provided by Oracle for Oracle Cloud Application Services implementations. 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. If you are not starting an Oracle Cloud Application Services implementation, sign into Oracle Identity Manager (OIM) as the OIM Administration users and provision the IT Security Manager job role with roles for user and role management. This enables the super user account, which is provisioned with the IT Security Manager job role, to create implementation users.

2. For starting all implementations, sign in as the user with initial access: either the Oracle Fusion Applications installation super user or the initial Oracle Cloud Application Services administrator user.

3. Select an offering to implement, and generate the setup tasks needed to implement the offering.

4. Perform the following security tasks:
   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.

5. 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.
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 for comparable Oracle Fusion applications 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.
- Achieve 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.

The following documentation resources are 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
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.
Define units of measure, unit of measure classes, and base units of measure for tracking, moving, storing, and counting items.

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

Unit of Measure Classes

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

Units of Measure

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

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

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

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

Scenario

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

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

Defining Unit of Measure Standard Conversions: Examples

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

This table lists examples of unit of measure classes, one unit of measure included in each class, the base unit of measure for the unit of measure class, and the conversion factor defined for the unit of measure.

<table>
<thead>
<tr>
<th>Unit of Measure Class</th>
<th>Unit of Measure</th>
<th>Base Unit of Measure</th>
<th>Conversion Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity</td>
<td>dozen</td>
<td>each</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(1 dozen = 12 each)</td>
</tr>
<tr>
<td>Weight</td>
<td>pound</td>
<td>gram</td>
<td>454</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(1 pound = 454 grams)</td>
</tr>
<tr>
<td>Time</td>
<td>minute</td>
<td>second</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(1 minute = 60 seconds)</td>
</tr>
</tbody>
</table>

FAQs for Manage Units of Measure

What’s a unit of measure standard conversion?

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

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

Preparing the IT Security Manager Job Role

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

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

- Identity User Administrators
- Role Administrators

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

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

Defining Initial Implementation Users

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

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

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

Creating Implementation Users

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

Note

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

Creating Data Roles for Implementation Users

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

Provisioning Roles to Implementation Users

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

**Caution**

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

---

**Initial Security Administration: Worked Example**

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

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

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

The following table summarizes key decisions for this scenario.

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

You create an initial implementation user by performing the following tasks.
1. The Oracle Identity Management System Administrator user provisions the IT Security Manager job role with roles for user and role management.

2. The Oracle Fusion Applications super user synchronizes LDAP users with HCM user management so that users can be provisioned with roles through HCM.

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

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

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

**Note**

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

---

### Preparing the IT Security Manager Role

The super user that was created when installing and provisioning Oracle Fusion Applications (for example, FAADMIN), or the initial administrator user provided by Oracle for Oracle Cloud Application Services, 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 Retrieve Latest LDAP Changes process takes some time to complete the first time it is run.

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

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

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

   The Oracle Identity Manager appears.

2. Click Create User.

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

3. Provide the following attributes:

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

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

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

Manage Currencies

Defining Currencies: Points to Consider

When creating or editing currencies, consider these points relevant to entering the currency code, date range, or symbol for the currency.

Currency Codes

You cannot change a currency code after you enable the currency, even if you later disable that currency.

Date Ranges

Users can enter transactions denominated in the currency only for the dates within the specified range. If you do not enter a start date, then the currency is valid immediately. If you do not enter an end date, then the currency is valid indefinitely.

Symbols

Even if you enter a symbol for a currency, the symbol is not always displayed when an amount is displayed in this currency. Some applications use currency symbols when displaying amounts. Others, like Oracle Fusion General Ledger, do not.

Manage Conversion Rate Types

Creating Conversion Rate Types: Critical Choices

Maintain different conversion rates between currencies for the same period with the Oracle Fusion General Ledger conversion rate types functionality. Four
predefined daily conversion rate types are seeded: Spot, Corporate, User, and Fixed, allowing you to use different rate types for different business needs. During journal entry, the conversion rate is provided automatically by the General Ledger based on the selected conversion rate type and currency, unless the rate type is user. For user rate types, you must enter the conversion rate. Define additional rate types as needed. Set your most frequently used rate type as the default. Conversion rate types cannot be deleted.

Assign conversion rate types to automatically populate the associated rate for your period average and period end rates for the ledger. For example, you can assign the predefined rate type Spot to populate your period average rates and the predefined rate type Corporate to populate your period end rates. Period average and period end rates are used in translation of account balances.

Conversion rate types are used to automatically assign a rate when you perform the following accounting functions:

- Convert foreign currency journal amounts to ledger currency equivalents
- Convert journal amounts from source ledgers to reporting currencies or secondary ledgers
- Run Revaluation or Translation processes

In creating new conversion rates, decide whether to do the following:

- Enforce inverse relationships
- Select pivot currencies
- Select contra currencies
- Enable cross rates and allow cross rate overrides
- Maintain cross rate rules

**Enforce Inverse Relationships**

Check the **Enforce Inverse Relationship** check box to specify whether or not to enforce the automatic calculation of inverse conversion rates when defining daily rates.

<table>
<thead>
<tr>
<th>Action</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checked</td>
<td>When you enter a daily rate to convert currency A to currency B, General Ledger automatically calculates the inverse rate, currency B to A, and enters it in the adjacent column. If either rate is changed, the application automatically recalculates the other rate. You can update the application calculated inverse rate, but once you do, the related rate is updated. The check box enforces that the inverse relationship is maintained but does not prevent you from updating the rates.</td>
</tr>
<tr>
<td>Unchecked</td>
<td>General Ledger calculates the inverse rate but you can change the rate and update the daily rates table without the corresponding rate being updated.</td>
</tr>
</tbody>
</table>
Select Pivot Currencies

Select a pivot currency that is commonly used in your currency conversions. A pivot currency is the central currency that interacts with contra currencies. For example, you set up a daily rate between the US dollar (USD) and the Euro currency (EUR) and another between the USD and the Canadian dollar (CAD). USD is the pivot currency in creating a rate between EUR and CAD. EUR and CAD are the contra currencies. Select the pivot currency from the list of values which contains those currencies that are enabled, effective, and not a statistical (STAT) currency. The description of the pivot currency is populated automatically based on the currency definition.

If you want the application to create cross rates against a base currency, define the base currency as the pivot currency. Selected pivot currencies can be changed in the Rate Types page.

Select Contra Currencies

Select currencies available on the list of values as contra currencies. The available currencies are those currencies which are enabled, effective, not STAT currency, and not the pivot currency selected earlier. The description of the contra currency is populated automatically based on the currency definition. Add or delete contra currencies in the Contra Currencies region of the Rate Types page.

Enable Cross Rates and Allow Cross Rate Overrides

Check the Enable Cross Rates check box to calculate conversion rates based on defined currency rate relationships. General Ledger calculates cross rates based on your defined cross rate rules. Associate your cross rate rules with a conversion rate type, pivot currency, and contra currencies. Cross rates facilitate the creation of daily rates by automatically creating the rates between contra currencies based on their relationship to a pivot currency. If the Enable Cross Rates check box is changed to unchecked after entering contra currencies, the application stops calculating cross rates going forward for that particular rate type. All the earlier calculated cross rates for that rate type remain in the database unless you manually delete them.

For example, if you have daily rates defined for the pivot currency, USD to the contra currency, EUR, and USD to another contra currency, CAD, the application will automatically create the rates between EUR to CAD and CAD to EUR. This prevents the need to manually define the EUR to CAD and CAD to EUR rates.

Check the Allow Cross Rates Override check box to permit your users to override application generated cross rates. If you accept the default of unchecked, the application generated cross rates cannot be overridden.

Maintain Cross Rate Rules

Define or update your cross rate rules at any time by adding or removing contra currency assignments. Add a contra currency to a cross rate rule and run
the Daily Rates Import and Calculation process to generate the new rates. If
your remove a cross rate rule or a contra currency from a rule, any cross rates
generated previously for that contra currency remain unless you manually delete
them. Changes to the rule are not retroactive and will not affect previously stored
cross rates. The Cross Rate process generates as many rates as possible and skips
currencies where one component of the set is missing.

Note

With a defined web service that extracts daily currency conversion rates
from external services, for example Reuters, currency conversion rates are
automatically updated for the daily rates and all cross currency relationships.

FAQs for Manage Conversion Rate Types

What's the difference between spot, corporate, user, and fixed rate types?

Spot, corporate, user, and fixed conversion rate types differ based on the
fluctuations of your entered foreign currency and your company procedures for
maintaining daily rates.

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

If you have infrequent foreign currency transactions, the user rate type can
simplify your currency maintenance while providing an accurate conversion rate
on the date of the transaction.
Define Enterprise Structures

Enterprise Structures: Overview

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

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

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

Legal Structure

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

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

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

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

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

**Management Structure**

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

**Functional Structure**

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

Enterprise Structures Business Process Model: Explained

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

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

**Note**

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

Global Enterprise Configuration: Points to Consider

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

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

Enterprise Configuration

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

Business Unit Management

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

Security Structure

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

Compliance Requirements

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

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

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

**Scenario**

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

**InFusion Corporation**

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

**Analysis**

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

- Your company is required to report using US Generally Accepted Accounting Principles (GAAP) standards and UK Statements of Standard Accounting Practice and Financial Reporting Standards. How many ledgers do you need to achieve proper statutory reporting?

- Your managers need reports that show profit and loss (revenue and expenses) for their lines of business. Do you use business units and balancing segments to represent your divisions and businesses? Do you secure data by two segments in your chart of accounts which represents each department and legal entity or one segment that represents both to produce useful, but confidential management reports?

- Your corporate management requires reports showing total organizational performance with drill down capability to the supporting details. Do you need multiple balancing segment hierarchies to achieve proper rollup of balances for reporting requirements?

- Your company has all administrative, account payables, procurement, and human resources functions performed at their corporate headquarters. Do you need one or more business unit in which to perform all these functions? How will your shared service center be configured?
Global Enterprise Structure Model

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

- Creation of three separate ledgers representing your separate legal entities:
  - InFusion America Inc.
  - InFusion Financial Services Inc.
  - InFusion UK Services Ltd.
- Consolidation of results for system components, installations, and maintenance product lines across the enterprise
- All UK general and administrative costs processed at the UK headquarters
- US Systems’ general and administrative costs processed at US Corporate headquarters
- US Financial Services maintains its own payables and receivables departments
In this chart, the green globe stands for mandatory and gold globe stands for optional setup. The following statements expand on the data in the chart.

- **The enterprise is mandatory because it serves as an umbrella for the entire implementation. All organizations are created within an enterprise.**

- **Legal entities are also mandatory. They can be optionally mapped to balancing segment values or represented by ledgers. Mapping balancing segment values to legal entities is mandatory if you plan to use the intercompany functionality.**

- **At least one ledger is mandatory in an implementation in which you record your accounting transactions.**

- **Business units are also mandatory because financial transactions are processed in business units.**

- **A shared service center is optional, but if used, must be a business unit.**

- **Divisions are optional and can be represented with a hierarchy of cost centers or by a second balancing segment value.**

- **Departments are mandatory because they track your employees.**

- **Optionally, add an item master organization and inventory organizations if you are tracking your inventory transactions in Oracle Fusion Applications.**

**Note**

Some Oracle Fusion Human Capital Management and Oracle Sales Cloud 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.
Essbase Character and Word Limitations

The following is a comprehensive list of character and word limitations that apply to Essbase. All of the limitations apply to all of the Oracle Fusion General Ledger configurations summarized in the table.

<table>
<thead>
<tr>
<th>Oracle Fusion General Ledger Configuration</th>
<th>Maps to Essbase As:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chart of Account Name</td>
<td>Cube Name</td>
</tr>
<tr>
<td>Chart of Account Segment Name</td>
<td>Dimension Name</td>
</tr>
<tr>
<td>Chart of Accounts Segment Value</td>
<td>Dimension Member Name</td>
</tr>
<tr>
<td>Chart of Accounts Segment Value Description</td>
<td>Alias for Member</td>
</tr>
<tr>
<td>Tree and Tree Version Name</td>
<td>Dimension Member Name</td>
</tr>
<tr>
<td>Primary Ledger Name</td>
<td>Dimension Member Name in Ledger Dimension</td>
</tr>
<tr>
<td>Secondary Ledger Name</td>
<td>Dimension Member Name in Ledger Dimension</td>
</tr>
<tr>
<td>Reporting Currency Name</td>
<td>Dimension Member Name in Ledger Dimension</td>
</tr>
<tr>
<td>Ledger Set Name</td>
<td>Dimension Member Name in Ledger Dimension</td>
</tr>
<tr>
<td>Accounting Calendar Period Names</td>
<td>Dimension Member Name in Accounting Period Name</td>
</tr>
<tr>
<td>Scenario Name Defined in Seeded Value Set Called Accounting Scenario</td>
<td>Dimension Member Name in Scenario Dimension</td>
</tr>
</tbody>
</table>

Even when case sensitivity is enabled in an aggregate storage outline for which duplicate member names is enabled, do not use matching names with only case differences for a dimension name. For example, do not:

- Name two dimensions Product and product.
- Use quotation marks or brackets.
- Use tabs in dimension, member, or alias names.
- Use accent characters.
- Use the characters for dimension or member names.

**Restricted Characters**

The following is a list of characters that are restricted and can not be used in dimension, member, or alias names.

<table>
<thead>
<tr>
<th>Character</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>at sign</td>
</tr>
<tr>
<td>\</td>
<td>backslash</td>
</tr>
<tr>
<td>,</td>
<td>comma</td>
</tr>
<tr>
<td>-</td>
<td>dash, hyphen, or minus sign</td>
</tr>
<tr>
<td>=</td>
<td>equal sign</td>
</tr>
<tr>
<td>&lt;</td>
<td>less than sign</td>
</tr>
<tr>
<td>()</td>
<td>parentheses</td>
</tr>
</tbody>
</table>
Other Restrictions

- Do not place spaces at the beginning or end of names. Essbase ignores such spaces.
- Do not use these types of words as dimension or member names:
  - Calculation script commands, operators, and keywords.
  - Report writer commands.
  - Function names and function arguments.
  - Names of other dimensions and members (unless the member is shared).
  - Generation names, level names, and aliases in the database.
- Any of these words in the table below:

<table>
<thead>
<tr>
<th>List 1</th>
<th>List 2</th>
<th>List 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>AND</td>
<td>ASSIGN</td>
</tr>
<tr>
<td>AVERAGE</td>
<td>CALC</td>
<td>CALCMBR</td>
</tr>
<tr>
<td>COPYFORWARD</td>
<td>CROSSDIM</td>
<td>CURMBRNAME</td>
</tr>
<tr>
<td>DIM</td>
<td>DIMNAME</td>
<td>DIV</td>
</tr>
<tr>
<td>DYNAMIC</td>
<td>EMPTYPARM</td>
<td>EQ</td>
</tr>
<tr>
<td>EQOP</td>
<td>EXCEPT</td>
<td>EXP</td>
</tr>
<tr>
<td>EXPERROR</td>
<td>FLOAT</td>
<td>FUNCTION</td>
</tr>
<tr>
<td>GE</td>
<td>GEN</td>
<td>GENRANGE</td>
</tr>
<tr>
<td>GROUP</td>
<td>GT</td>
<td>ID</td>
</tr>
<tr>
<td>IDERROR</td>
<td>INTEGER</td>
<td>LE</td>
</tr>
<tr>
<td>LEVELRANGE</td>
<td>LOOPBLOCK</td>
<td>LOOPPARMS</td>
</tr>
<tr>
<td>LT</td>
<td>MBR</td>
<td>MBRNAME</td>
</tr>
<tr>
<td>MBRONLY</td>
<td>MINUS</td>
<td>MISSING, #MISSING</td>
</tr>
<tr>
<td>MUL</td>
<td>MULOP</td>
<td>NE</td>
</tr>
<tr>
<td>NON</td>
<td>NONINPUT</td>
<td>NOT</td>
</tr>
<tr>
<td>OR</td>
<td>PAREN</td>
<td>PARENPARM</td>
</tr>
<tr>
<td>PERCENT</td>
<td>PLUS</td>
<td>RELOP</td>
</tr>
<tr>
<td>SET</td>
<td>SKIPBOTH</td>
<td>SKIPMISSING</td>
</tr>
<tr>
<td>SKIPNONE</td>
<td>SKIPZERO</td>
<td>TO</td>
</tr>
<tr>
<td>TOLOCALRATE</td>
<td>TRAILMISSING</td>
<td>TRAILSUM</td>
</tr>
</tbody>
</table>
Define Initial Configuration with the Enterprise Structures Configurator

Establishing Enterprise Structures Using the Enterprise Structures Configurator: Explained

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

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

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

**Establish Enterprise Structures**

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

**Establish Job and Position Structures**

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

**Review Configuration**

You can view a result of the interview process prior to loading the configuration. In the review results, you can view the divisions, legal entities, business units, reference data sets, and the management reporting structure that the application will create when you load the configuration.

**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.

**Rolling Back an Enterprise Structure Configuration: Explained**

The Enterprise Structures Configurator (ESC) provides the ability to roll back an enterprise configuration in the following circumstances:

**Manual Rollback**

You can manually roll back an enterprise configuration after loading it, for example, because you decide you do not want to use it. Clicking the Roll Back Configuration button on the Manage Enterprise Configuration page rolls back any enterprise structures that were created as a part of loading the configuration.
Automatic Rollback

If an error occurs during the process of loading the configuration, then the application automatically rolls back any enterprise structures that were created before the error was encountered.

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.

Resulting Enterprise Configuration

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

**Divisions**

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

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

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

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

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

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

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

**Legal Entities: Explained**

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

Legal entities have the right to own property, the right to trade, the responsibility to repay debt, and the responsibility to account for themselves to regulators, taxation authorities, and owners according to rules specified in the relevant legislation. Their rights and responsibilities may be enforced through the judicial system. Define a legal entity for each registered company or other entity recognized in law for which you want to record assets, liabilities, expenses and income, pay transaction taxes, or perform intercompany trading.

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

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

**The Role of Your Legal Entities**

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

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

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

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

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

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

Using the Enterprise Structures Configurator (ESC), you can create legal entities for your enterprise automatically, based on the countries in which divisions of your business operate, or you can upload a list of legal entities from a spreadsheet.

Automatically Creating Legal Entities

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

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

**Example: Creating Legal Entities Automatically**

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

This figure illustrates InFusion Corporation’s enterprise structure.

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

<table>
<thead>
<tr>
<th>Country</th>
<th>Enterprise</th>
<th>InFusion Lighting</th>
<th>InFusion Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Japan</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>US</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>UK</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>India</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Business Unit Level</th>
<th>Suggested Business Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
<td>• US</td>
</tr>
<tr>
<td></td>
<td>• UK</td>
</tr>
<tr>
<td></td>
<td>• Japan</td>
</tr>
<tr>
<td></td>
<td>• India</td>
</tr>
<tr>
<td>Country and Division</td>
<td>• InFusion Lighting: Japan</td>
</tr>
<tr>
<td></td>
<td>• InFusion Lighting: US</td>
</tr>
<tr>
<td></td>
<td>• Infusion Security: UK</td>
</tr>
<tr>
<td></td>
<td>• Infusion Security: India</td>
</tr>
<tr>
<td>Country and business function</td>
<td>• Sales: Japan</td>
</tr>
<tr>
<td></td>
<td>• Marketing: Japan</td>
</tr>
<tr>
<td></td>
<td>• Sales: US</td>
</tr>
<tr>
<td></td>
<td>• Sales: UK</td>
</tr>
<tr>
<td></td>
<td>• Marketing: India</td>
</tr>
<tr>
<td></td>
<td>• Sales: India</td>
</tr>
<tr>
<td>Division</td>
<td>• InFusion Lighting</td>
</tr>
<tr>
<td></td>
<td>• InFusion Security</td>
</tr>
<tr>
<td>Division and Legal Entity</td>
<td>• InFusion Lighting: Japan</td>
</tr>
<tr>
<td></td>
<td>• InFusion Lighting: US</td>
</tr>
<tr>
<td></td>
<td>• Infusion Security: UK</td>
</tr>
<tr>
<td></td>
<td>• Infusion Security: India</td>
</tr>
</tbody>
</table>
### 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

---

| Division and Business Function | • InFusion Lighting, Sales  
|                               | • InFusion Lighting, Marketing  
|                               | • InFusion Security, Sales  
|                               | • InFusion Security, Marketing  
| Business Function              | • Sales  
|                               | • Marketing  
| Legal Entity                  | • Legal Entity: Japan  
|                               | • Legal Entity: US  
|                               | • Legal Entity: UK  
|                               | • Legal Entity India  
| Legal Entity and Business Function | • Legal Entity: Japan, Sales  
|                                  | • Legal Entity: Japan, Marketing  
|                                   | • Legal Entity: US, Sales  
|                                    | • Legal Entity: UK, Sales  
|                                     | • Legal Entity India, Marketing  
|                                      | • Legal Entity India, Sales  

---
all business units or application components using that set. You can assign a separate set to each business unit for the type of object that is being shared. For example, assign separate sets for payment terms, transaction types, and sales methods to your business units.

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

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

Reference Data Sharing Methods

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

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

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

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

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

**Common Set Versus Specific Sets**

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

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

**Business Unit Set Assignment**

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

**Example: Assigning Sets to Business Units**

InFusion Corporation has two divisions: Lighting and Security, and the divisions each have two locations. Each location has one or more business functions.

The following figure illustrates the structure of InFusion Corporation.
When deciding how to create business units, InFusion decides to create them using the country and business function level. Therefore, they created the following business units:

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

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

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

InFusion assigns business units to sets as follows:

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

When setting up grades, departments, and locations for the business units, InFusion will assign the data to the default set for each business unit. When setting up jobs, they will assign the Jobs set and will assign the Common Set to any jobs that may be used throughout the entire organization.

When using grades, departments, and locations at the transaction level, users will be able to select data from the set that corresponds to the business unit that
they enter on the transaction, and any data that was assigned to the Common Set. For example, for transactions for the Marketing_Japan business unit, grades, locations, and departments from the Mktg_Japan_Set will be available to select, as well as from the Common Set.

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

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

Creating Reference Data Sets in the Enterprise Structures Configurator: Explained

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

A standard set called the Enterprise set is predefined.

**Common Set**

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

**Jobs and Positions: Critical Choices**

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

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

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

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

**Primary Industry of Your Enterprise**

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

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

**Management of People**

The following table displays suggestions of whether to use jobs or a combination of jobs and positions based on your industry and how you manage your employees when there is turnover.
Define Enterprise Structures 6-31

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

**Positions: Examples**

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

**Retail Industry**

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

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

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

This figure illustrates the retail position setup.
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](image)

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.
Define Enterprise Structures

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 one
corporate business unit specific set. At transaction entry, the list of values for payment terms consists of only one set of data; the set that is assigned to the transaction’s business unit.

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

Assigning Reference Data Sets to Reference Objects: Points to Consider

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

Determinant Types

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

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

Determinant

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

Reference Groups

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

**Note**

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

---

**Items and Supplier Site Reference Data Sharing: Explained**

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

Items 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 on 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.

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

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

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

Enterprise: Explained

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

Enterprise Defined

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

An enterprise organization captures the name of the deploying enterprise and the location of the headquarters. There is normally a single enterprise organization in a production environment. Multiple enterprises are defined when the system is used to administer multiple customer companies, for example, multiple tenants, or when a customer chooses to set up additional enterprises for testing or development.

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

What reference data objects can be shared across cost organizations?

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

<table>
<thead>
<tr>
<th>Application Name</th>
<th>Reference Data Object</th>
<th>Method of Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Assignment to one set only, with common values</td>
</tr>
</tbody>
</table>

What reference data objects can be shared across project units?

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

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

<table>
<thead>
<tr>
<th>Application Name</th>
<th>Reference Data Object</th>
<th>Method of Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Assignment to multiple sets, no common values allowed</td>
</tr>
</tbody>
</table>
Oracle Fusion Applications offers capabilities for multiple tenants to share the same applications instance for some human resources processes. If you offer business process outsourcing services to a set of clients, each of those clients may be represented as an enterprise within an Oracle Fusion Application instance. To support this functionality, system owned reference data such as sequences, sets, and flexfields are also defined within an enterprise.

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

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

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

**Define Enterprise: Manage Locations**

**Locations: Explained**

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

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

**Location Sets**

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

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

The following figure shows how locations sets restrict access to users.
Uploading Locations Using a Spreadsheet

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, add your location information to the spreadsheet, and then upload directly to your enterprise configuration. You can upload the spreadsheet multiple times to accommodate revisions.

FAQs for Manage Locations

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

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

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

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

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

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

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

How can I associate a location with an inventory organization?

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

Define Geographies

Defining Address Cleansing: Explained

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

To be able to use the address cleansing functionality, you need to have license for the customer data quality application, because the feature is delivered using data quality integration.

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 the application does not find a matching address it 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 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 comprises all geography related data, including geography types and geographies.

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:

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

**Importing Country Structures Using File-Based Import: Explained**

This topic explains how to prepare and import country structure data from an external data source into Oracle Sales Cloud using the File-Based Data Import feature. A country structure is a hierarchical grouping of geography types for a country. For example, the geography structure for the United States has the geography type of State at the top, followed by the County, then the City, and finally the Postal Code.

You can use the country structure to set up the following:

- The relationships between geographies within a country
- The types of geographies that you can define for a country

Consider the following questions when importing your data:

- How does your legacy system or source system represent the country structure compared to how Oracle Sales Cloud represents the same data?
- Do you have to configure values in Oracle Sales Cloud to map to your data values?
- Do you have to customize Oracle Sales Cloud to capture additional attributes that are critical to the way you do business?
- What import features are available for importing your business object?
- How do you verify your imported data?

**Comparing Business Object Structures**

You must understand how your country structure data corresponds with the data in Oracle Sales Cloud in order to be able to map your legacy data to the data needed by Oracle Sales Cloud. First, you must understand how Oracle Sales Cloud represents the structure of the data for a country structure.

You must import a separate country structure import object for each country. Each of these import objects must contain the geography types that are used in the country's structure, organized in a hierarchy using geography level numbers. For example, if you are importing the country structure of Australia, the country structure could be the following: 1: Country, 2: State, 3: County, 4: Town, 5: ZIP.
Import Objects for the Country Structure

To facilitate the import of country structures, Oracle Sales Cloud incorporates the structure of the country structure into import objects. The import object for country structures is GeoStructureLevel.

Comparing Business Object Data

Each import object is a collection of attributes that helps to map your data to the Oracle Sales Cloud data and to support one-to-many relationships between the structural components that make up the country structure.

A good understanding of the attribute details of the import objects is critical to preparing your import data. For information about the Oracle Sales Cloud attributes, see the Oracle Enterprise Repository. The reference files contain descriptions, logic used to choose default values, and validation information for each of the Oracle Sales Cloud attributes. The validation information includes the navigation to the task where you can define values in Oracle Sales Cloud. For example, if you have values in your data that correlate to a choice list in Oracle Sales Cloud, then the validation information for that attribute provides the task name in the Setup and Maintenance work area where you can define your values. For additional information, including a list of reference file names and locations that you need to complete this task, see the following table.

<table>
<thead>
<tr>
<th>Import Object</th>
<th>Related Import Object Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country Structure</td>
<td>Country Structure Import Objects: How They Work Together</td>
</tr>
</tbody>
</table>

Extensible Attributes

If you need to extend the Oracle Sales Cloud object to import your legacy or source data, you must use Application Composer to design your object model extensions and to generate the required artifacts to register your extensions and make them available for importing. The corresponding import object is updated with the extensible attributes, which can then be mapped to your source file data. You can use the same source file to import both extensible custom attributes and the standard import object attributes.

Importing Country Structures Using File-Based Data Import

For the country structure business object, you must use the File-Based Data Import feature. You prepare XML or text source data files in a form that is suitable for file-based import. The file-based import process reads the data included in your source file, populates the interface tables according to your mapping, and imports the data into the application destination tables.

The Define File-Based Data Import Setup and Maintenance task list includes the tasks needed to configure the import objects, to create source-file mappings, and to schedule the import activities. You submit file-based import activities for each import object. When creating a new country structure, you import the Country Structure object.
You must be assigned the Master Data Management Administrator job role to access and submit the import activities for country structures.

**Verifying Your Imported Data**

You can view the list of import activities from the Manage Import Activities page. You can verify your imported data by clicking the Status column for your import activity.

**Country Structure Import Objects: How They Work Together**

This topic describes the Country Structure import object. You use the Country Structure import object when you submit a file-based import activity to import your country structure information. This topic introduces the following:

- Target import object concepts
- Target objects for the Country Structure import object
- Target import object attributes
- Target object attribute reference guide files

**Target Import Object Concepts**

The Country Structure import object is used to import a country structure hierarchy, including details, such as geography type, geography type name, parent geography type, geography level numbers, and so on. To map the source data in your import file to the target attributes in Oracle Sales Cloud, you must understand how the target objects are related and what attributes are included in each target object.

**Country Structure Target Import Objects**

The Country Structure import object contains one target import object that organizes the individual attributes of the different aspects of the geography structure. When updating an existing country structure, you must provide the parent reference information of the existing country structure. This reference information connects the imported geography structure to the existing one. Use the ImpGeoStructureLevel target import object to create and update country structure information.

**Target Import Objects Attributes**

You must compare the attributes that you want to import with the target object attributes that are available and their valid values. To evaluate your source data and Oracle Sales Cloud attributes for mapping and validation, you use an Oracle Enterprise Repository reference guide, which is available for each target import object. The reference guide file includes attribute descriptions, default values, and validations performed by the import process. Review the validation for each attribute to determine whether there are functional prerequisites or setup tasks that are required.

To import your source file data, you define a mapping between your source file data and the combination of the target object and target object attribute. You can predefine and manage import mappings using the File-Based Import Mapping task, or you can define the mapping when you define the import activity using...
the File-Based Import Activity task. Both tasks are available in the Setup and Maintenance work area.

**Note**
If any of the attributes you want to import does not have an equivalent target object attribute, then review the Application Composer extensibility features for country structures.

**Target Import Objects Attributes Resources**
To access the reference guide files for the country code’s target import objects, see the File-Based Data Import assets in Oracle Enterprise Repository (http://fusionappsoer.oracle.com).
For detailed information on importing geographies using file-based import, refer to Document No. 1481758.1, Importing Master Reference Geography Data, on the Oracle Support site.

The following table lists the reference guide files that are available from the Documentation tab for the Country Code File-Based Data Import asset.

<table>
<thead>
<tr>
<th>Target Import Object</th>
<th>Description</th>
<th>Reference Guide File Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImpGeoStructureLevel</td>
<td>Contains information that specifies a country’s geography structure.</td>
<td>HZ_IMP_GEO_STRUCTURE _LEVELS_Reference</td>
</tr>
<tr>
<td></td>
<td>Sample attributes: GeographyType, GeographyTypeName, LevelNumber, and ParentGeographyType.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reference attribute: CountryCode</td>
<td></td>
</tr>
</tbody>
</table>

**Importing Geographies Using File-Based Import: Explained**

This topic describes the tasks you must perform to import geography information. A geography is any region with a boundary around it, regardless of its size. It might be a state, a country, a city, a county, or a ward. You must create or import geographies before you can associate them with custom zones and addresses.

Consider the following questions when importing your data:
- How does your legacy system or source system represent the geography compared to how Oracle Sales Cloud represents the same data?
- Do you have to configure values in Oracle Sales Cloud to map to your data values?
- What import features are available for importing your business object?
- How do you verify your imported data?

**Comparing Business Object Structures**

You must understand how your geography data corresponds with the data in Oracle Sales Cloud in order to be able to map your legacy data to the data.
needed by Oracle Sales Cloud. First, you must understand how Oracle Sales Cloud represents the structure of the data for a geography.

You must import a separate country structure import object for each country. Each of these import objects must contain the geography types that are used in the country’s structure, organized in a hierarchy using geography level numbers. For example, if you are importing the country structure of Australia, the country structure could be the following: 1: Country, 2: State, 3: County, 4: Town, 5: ZIP.

**Import Objects for the Geography**

To facilitate the import of geographies, Oracle Sales Cloud incorporates the structure of the geography into import objects. The import object for the geography is ImpGeography.

**Comparing Business Object Data**

Each import object is a collection of attributes that helps to map your data to the Oracle Sales Cloud data and to support one-to-many relationships between the structural components that make up the geography. A good understanding of the attribute details of the import objects is critical to preparing your import data. For information about the Oracle Sales Cloud attributes, see the Oracle Enterprise Repository. The reference guide files contain descriptions, logic used to choose default values, and validation information for each import object attribute. The validation information includes the navigation to the task where you can define values in Oracle Sales Cloud. For example, if you have values in your data that correlate to a choice list in Oracle Sales Cloud, then the validation information for that attribute provides the task name in the Setup and Maintenance work area where you can define your values. For additional information, including a list of reference file names and locations that you need to complete this task, see the following table.

<table>
<thead>
<tr>
<th>Import Object</th>
<th>Related Import Object Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImpGeography</td>
<td>Geography Import Objects: How They Work Together</td>
</tr>
</tbody>
</table>

Hint: You can use the keyword importing geographies to search for related topics in Help.

**Extensible Attributes**

Oracle Sales Cloud does not support extensible attributes for geographies. You can only import data for attributes provided by Oracle Sales Cloud.

**Importing Geographies Using File-Based Data Import**

For the geography business object, you must use the File-Based Data Import feature. You prepare XML or text source data files in a form that is suitable for file-based import. The file-based import process reads the data included in your source file, populates the interface tables according to your mapping, and imports the data into the application destination tables.

The Define File-Based Data Import Setup and Maintenance task list includes the tasks needed to configure the import objects, to create source-file mappings,
and to schedule the import activities. You submit file-based import activities for each import object. When creating a new geography, you import the Geography object. You must be assigned the Master Data Management Administrator job role to access and submit the import activities for geographies.

When importing geography information, you must provide the parent reference information for all parent levels for the entity.

**Verifying Your Imported Data**

Oracle Sales Cloud provides File-Based Import activity reports, which can be used to verify imported data. Users with the Master Data Management Administrator job role can also navigate to the Manage Geographies work area to view the imported geographies.

**Geography Import Objects: How They Work Together**

This topic describes the Geography import object. You use the Geography import object to import geography information.

This topic introduces the following:

- Target import object concepts
- Target objects for the Geography import object
- Target import object attributes
- Target import object attribute reference guide files

**Target Import Object Concepts**

The Geography import object is used to import geography hierarchy information to create or update the geography data of a country. To map the source data in your import file to the target attributes in Oracle Sales Cloud, you must understand how the target objects are related and what attributes are included in each target object.

**Geography Target Import Objects**

The target import objects in the Geography import object contain information about the geography hierarchy. When updating an existing geography, you must provide the parent reference information of the existing geography, which connects the geography to the country of which it is a part.

Use the ImpGeography target import object to create and update geography information.

**Note**

Before you import geography data for a country, you must define the country’s geography structure.

**Target Import Objects Attributes**

You must compare the attributes that you want to import with the target object attributes that are available and their valid values. To evaluate your source data and Oracle Sales Cloud attributes for mapping and validation, you use an Oracle
Enterprise Repository reference guide, which is available for each target import object. The reference guide file includes attribute descriptions, default values, and validations performed by the import process. Review the validation for each attribute to determine whether there are functional prerequisites or setup tasks that are required.

To import your source file data, you define a mapping between your source file data and the combination of the target object and target object attribute. You can predefine and manage import mappings using the File-Based Import Mapping task, or you can define the mapping when you define the import activity using the File-Based Import Activity task. Both tasks are available in the Setup and Maintenance work area.

**Target Import Objects Attributes Resources**

To access the reference guide files for the geography’s target import objects, see the File-Based Data Import assets in Oracle Enterprise Repository (http://fusionappsor.oracle.com).

For detailed information on importing geographies using file-based import, refer to Document No. 1481758.1, Importing Master Reference Geography Data, on the Oracle Support site.

The following table lists the reference guide files that are available from the Documentation tab for the Geography File-Based Data Import asset.

<table>
<thead>
<tr>
<th>Target Import Object</th>
<th>Description</th>
<th>Attribute Reference Guide File Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImpGeography</td>
<td>Contains information that captures a country’s geography hierarchy details. Sample attributes: CountryCode, GeoDataProvider, GeographyType, PrimaryGeographyCode, PrimaryGeographyCodeType, and PrimaryGeographyName. Reference attribute: CountryCode</td>
<td>HZ_IMP_GEOGRAPHIES_T_Reference</td>
</tr>
</tbody>
</table>

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

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

The following table summarizes the key decisions that you need to make in this scenario:

<table>
<thead>
<tr>
<th>Decisions to Consider</th>
<th>In This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>What type of object are you importing?</td>
<td>Geography</td>
</tr>
<tr>
<td>What file type are you using for your source data?</td>
<td>Text file</td>
</tr>
</tbody>
</table>
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 Sales Cloud? | All, except for the RecordTypeCode field
When do you want to process the import? | Immediately

**Summary of the Tasks**

These are the steps that are required to create an import activity and activate 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 be longer than 18 characters. you could structure your geography source data as follows:

<table>
<thead>
<tr>
<th>Geography Level</th>
<th>Name</th>
<th>Source ID</th>
<th>Parent Source ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Country)</td>
<td>US</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2 (State)</td>
<td>CA</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>3 (County)</td>
<td>Alameda</td>
<td>111</td>
<td>11</td>
</tr>
<tr>
<td>4 (City)</td>
<td>Pleasanton</td>
<td>1111</td>
<td>111</td>
</tr>
</tbody>
</table>
Create and Schedule the Import Activity

You create an import activity, enter the import details, and schedule the import. An import activity definition provides instructions for the import processing, including details related to selecting the source file, or file location; mapping fields from the source file to the Oracle Sales Cloud database object and attribute; and scheduling the import.

1. Navigate to Setup and Maintenance and search for the Manage File Import Activities task. Click Go to Task.

2. In the Manage Import Activities page, click the Create icon.

3. In the Create Import Activity: Set Up page, create an import activity for the Geography object type by completing the fields, as shown in this table:

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

**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 Sales Cloud database object and attribute, as shown in this example:

<table>
<thead>
<tr>
<th>Column Header</th>
<th>Example Value</th>
<th>Ignore</th>
<th>Object</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Geography</td>
<td>Name</td>
<td>United</td>
<td>Imp</td>
<td>Primary Geography</td>
</tr>
<tr>
<td>Name</td>
<td>Primary</td>
<td>States</td>
<td>Imp</td>
<td>Name</td>
</tr>
<tr>
<td>Country Code</td>
<td>US</td>
<td>No</td>
<td>Imp</td>
<td>Country Code</td>
</tr>
<tr>
<td>Record Type Code</td>
<td>0</td>
<td>Yes</td>
<td>Imp</td>
<td>Record Type Code</td>
</tr>
<tr>
<td>Source ID</td>
<td>10265</td>
<td>No</td>
<td>Imp</td>
<td>Source ID</td>
</tr>
</tbody>
</table>
If you do not want to import a column in the text file you can select Ignore.

Note
If you have any difficulties mapping the fields from your source file to the relevant Oracle Sales Cloud database 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 as soon as you activate it.

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, verify your import details in the Import Details, File Details, Import Options, and Schedule sections. Update the import details if required by navigating to the previous screens using the Back link.

2. Once you are sure your import details are correct, 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 Sales Cloud applications instance, and then after the territory geography zones are defined you can export the territory zones and import them into another Oracle Sales Cloud instance.

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

1. Import the master reference geography data into the Oracle Sales Cloud.

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 Sales Cloud 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 Sales Cloud 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 Sales Cloud instance before you import the configuration package.

---

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

This example shows how to configure the geography structure, hierarchy, and validation for a country geography, using the United Kingdom country geography as an illustration.

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

Add the County and Post Town geography types to the geography structure. Next, 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**

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 **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**

To begin creating the geography hierarchy for the United Kingdom, 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. Define the geography mapping and validation for the United Kingdom default address style format. Then map the geography types to attributes, enable the geography types for Lists of Values and Geography validation, and set the geography validation level.

1. On the Manage Geographies page, click **Validation Defined**.

2. On the Manage Geography Validation page, Address Style section, click **No Styles Format** to highlight the table row.

3. For the County geography type, click the **County** list item in the **Map to Attribute** field.

4. Click the **Enable List of Values** option for the County geography type.

5. Click the **Geography Validation** option for the County geography type.

6. For the Post Town geography type, click the **City** list item in the **Map to Attribute** field.

7. Click the **Geography Validation** option for the Post Town geography type.

8. In the Geography Validation Control section, click the **Error** list item in the **Geography Validation Level for Country** field.

9. Click **Save and Close**.

**FAQs for Define Geographies**

**When do I define address cleansing?**

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

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

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

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

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

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

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

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

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

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

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

**Define Legal Entity**

**Legal Entities: Explained**

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

Legal entities have the right to own property, the right to trade, the responsibility to repay debt, and the responsibility to account for themselves to regulators, taxation authorities, and owners according to rules specified in the relevant legislation. Their rights and responsibilities may be enforced through the judicial system. Define a legal entity for each registered company or other entity recognized in law for which you want to record assets, liabilities, expenses and income, pay transaction taxes, or perform intercompany trading.

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

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

The Role of Your Legal Entities

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

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

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

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

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

Legal Entity in Oracle Fusion: Points to Consider

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

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

**The Importance of Legal Entity in Transactions**

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

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

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

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

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

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

**Legal Entity and Its Relationship to Divisions**

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

**Legal Entity and Its Relationship to Ledgers**

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

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

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

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

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

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

**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: Manage Chart of Accounts Structures and Structure Instances

Chart of Accounts: Explained

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

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

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

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

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

Thick Versus Thin General Ledger: Critical Choices

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

- A general ledger used in conjunction with an enterprise profitability management (EPM) product, which has data standardized from each operation, is designed as a thin general ledger. Use this variation if your solution is project based, and Oracle Fusion Projects is implemented. More detailed reporting can be obtained from the Projects system. In the thin general ledger, business units, divisions, and individual departments are not represented in the chart of accounts.

- A general ledger, with segments representing all aspects and capturing every detail of your business, with frequent posting, many values in each segment, and many segments, is called a thick general ledger. A thick general ledger is designed to serve as a repository of management data for a certain level of management. For example, a subsidiary’s general ledger is designed to provide the upper management enough data to supervise operations, such as daily sales, without invoice details or inventory without part number details.

- A primary ledger and a secondary ledger, where one is a thick general ledger and the other a thin general ledger, provides dual representation for reporting requirements that require more than one ledger.

**Thin General Ledger**

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

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

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

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

With a thick general ledger, you use the general ledger as a detailed, analytic tool, performing analytic functions directly in the general ledger. Data is broken down by many reporting labels, and populated frequently from the subledgers. You maximize the data stored in the general ledger. A thick general ledger has many of the following characteristics:

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

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

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

Other Considerations

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

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

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

- Minimal disclosure to the authorities in addition to the requirements listed above. For example, in some European countries, fiscal authorities examine ledgers at the detailed account level.
- Fiscal only adjustments, allocations, and revaluations, which don't impact the thick general ledger.

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

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

This decision affects the amount of:

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

Note

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

Summary

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

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

Chart of Accounts: How Its Components Fit Together

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

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

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

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

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

**Segment Labels**

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

- **Balancing**: Ensures that all journals balance for each balancing segment value or combination of multiple balancing segment values to use in trial balance reporting. There are three balancing segment labels: primary, second, and third balancing. The primary balancing segment label is required.

- **Cost Center**: Facilitates grouping of natural accounts by functional cost types, accommodating tracking of specific business expenses across natural accounts. As cost centers combine expenses and headcount data into costs, they are useful for detailed analysis and reporting. Cost centers are optional, but required if you are accounting for depreciation, additions, and other transactions in Oracle Fusion Assets, and for storing expense approval limits in Oracle Fusion Expense Management.

- **Natural Account**: Determines the account type (asset, liability, expense, revenue, or equity) and other information specific to the segment value. The natural account segment label is required.

- **Management**: Optionally, denotes the segment that has management responsibility, such as the department, cost center, or line of business. Also can be attached to the same segment as one of the balancing segments to make legal entity reporting more granular.

- **Intercompany**: Optionally, assigns the segment to be used in intercompany balancing functionality.

**Note**

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

**Account Combinations**

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

**Rules**

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

- **Security rules**: Prohibit certain users from accessing specific segment values. For example, you can create a security rule that grants a user access only to his or her department.

- **Cross-validation rules**: Control the account combinations that can be created during data entry. For example, you may decide that sales cost centers 600 to 699 should enter amounts only to product sales accounts 4000 to 4999.
Create Chart of Accounts, Ledger, Legal Entities, and Business Units in Spreadsheets: Explained

Represent your enterprise structures in your chart of accounts, ledger, legal entities, and business unit configuration to track and report on your financial objectives and meet your reporting requirements. These components are the underlying structure for organizing financial information and reporting.

The chart of accounts within the ledger facilitates aggregating data from different operations, from within an operation, and from different business flows. This functionality enables you to report using consistent definitions to your stakeholders in compliance with legislative and corporate reporting standards and aids in management decisions.

Rapid implementation is a way to configure the Oracle Fusion Financial Enterprise and Financial Reporting Structures quickly using sheets in a workbook to upload lists of companies (legal entities), ledgers, business units, chart of account values, and other similar data. Once the sheets have been uploaded, the application creates your ledger, business unit, and other components. The following graphic shows the relationship of these components.

- **Legal Entities**: Identifies a recognized party with rights and responsibilities given by legislation, which has the right to own property and the responsibility to account for themselves.
- **Chart of Accounts**: Configures accounts consisting of components called segments that are used to record balances and organize your financial information and reporting.
- Segments: Contains a value set that provides formatting and validation of the set of values used with that segment. When combined, several segments create an account for recording your transactions and journal entries.

- Segment Labels: Identifies certain segments in your chart of accounts and assigns special functionality to those segments. The three required segment labels are:
  - Balancing Segment: Ensures that all journals balance for each balancing segment value or combination of multiple balancing segment values to use in financial processes and reporting. The three balancing segment labels are: primary, second, and third balancing. The primary balancing segment label is required.
  - Natural Account: Facilities processes in the General Ledger application, such as retained earnings posting. Determines the account type, which includes asset, liability, expense, revenue, or equity.
  - Cost Center: Facilitates grouping of natural accounts by functional cost types, accommodating tracking of specific business expenses across natural accounts.

- Ledger: Maintains the records and is a required component in your configuration. The Rapid implementation process:
  - Creates your ledger by combining your chart of accounts, calendar, and currency as well as other required options defined in the sheets.
  - Assigns a default for the fourth component, the subledger accounting method, used to group subledger journal entry rule sets together to define a consistent accounting treatment.
  - Creates a balances cube for each ledger with a unique chart of accounts and calendar. Each segment is created as a dimension in the balances cube.
  - Business Units with Business Functions: Identifies where subledger transactions are posted and provides access to perform subledger business processes. Business units are assigned to a primary ledger, as well as a default legal entity, when configured and identify where subledger transactions are posted.
  - Subledgers: Captures detailed transactional information, such as supplier invoices, customer payments, and asset acquisitions. Uses subledger accounting to transfer transactional balances to the ledger where they are posted.

Create Chart of Accounts, Ledger, Legal Entities, and Business Units in Spreadsheets: How They Are Processed

The Create Chart of Accounts, Ledger, Legal Entities, and Business Units rapid implementation process consists of four steps.

1. Enter the data into the sheets.
2. Upload the XML files generated from the sheets.
3. Run the deployment process to finalize the chart of accounts configuration.
4. Upload the XML files generated from the sheets for the rest of the configuration.

Note
On the Instruction sheet is a link to a completed sample data workbook.

Process Overview
Begin by downloading the Rapid Implementation for General Ledger workbook using the Create Chart of Accounts, Ledger, Legal Entities, and Business Units in Spreadsheet task on the Setup and Maintenance work area.

The following figure illustrates the Create Chart of Accounts, Ledger, Legal Entities, and Business Units process, what data is entered into each sheet of the workbook, and the components that the process creates.

**Process**
**Enter Data**
The Create Chart of Accounts, Ledger, Legal Entities, and Business Units workbook provides five sheets.

1. Instructions
2. Chart of Accounts, Calendar, and Ledger
3. Business Units
4. Companies and Legal Entities
5. Natural Accounts

Sheets used to enter other segment values and hierarchies for additional segments are created by entering the segments on the Chart of Accounts, Calendar, and Ledger sheet and then clicking the Add Segment Sheets button.

Instructions Sheet

Read the planning tips, loading process, best practices, and recommendations.

Chart of Accounts, Calendar, and Ledger Sheet

Enter your data to create your ledger, its components, chart of accounts, currency, and calendar, and set the required ledger options.

- **Ledger** name is the name of your primary ledger and often appears in report titles, so enter a printable name.

- **Ledger Currency** represents the currency that most of your transactions are entered.

- **Retained Earnings Account** is used when you open the first period of a new year. The application moves the total balances in your revenue and expense accounts to the Retained Earnings accounts by balancing segment.

**Tip**

When the data is uploaded, the Allow Dynamic Insertion option used to enable the generation of new account combinations dynamically instead of creating them manually is enabled by default. To prevent the creation of invalid accounts, you must define cross-validation rules. Define cross-validation rules before entering data or loading history. Cross-validation rules only prevent creation of new accounts, not disabling of preexisting accounts.
• **Enable Average Balances** is used to enable Average Balances functionality.

The Average Balance feature provides organizations with the ability to track average and end-of-day balances, report average balance sheets, and create custom reports using both standard and average balances. Average balance processing is important for financial institutions, since average balance sheets are required, in addition to standard balance sheets, by many regulatory agencies. Many organizations also use average balances for internal management reporting and profitability analysis.

---

**Tip**

If you select No and uploaded the options, this region cannot be changed and does not display on the Specify Ledger Options page.

---

• **Fiscal Year Start Date** is the beginning date of your calendar for the ledger and cannot be changed once the ledger is saved.

---

**Important**

Select a period before the first period you plan to load history or perform translations to enable running translation. You cannot run translation in the first defined period of a ledger calendar.

---

• **Period Frequency** must be Monthly and is predefined.

---

**Note**

If you require a calendar other than monthly, such as 4-4-5 or weekly, define the calendar in the regular calendar page.

---

• **Adjusting Periods** add one or more periods that are used to enter closing, audit, or other adjustments in the General Ledger at quarter or year end. The entries are tracked in the adjusting period and not in your monthly activity.

• **Chart of Accounts** region is where you enter your segments, segment labels, short prompts, and display length data that is used to create your chart of accounts. Plan this data carefully, as you are defining the basic structure for your accounting and reporting.

• **Display Length** sets the segment size so select carefully and leave room for growth. For example, if you have 89 cost centers, enter 3 for the Display Length to allow for more than 100 cost centers in the future.

• **Add Segment Sheets** button to create sheets for additional segments. Only the Company and Natural Account segment sheets are provided.

---

**Note**

If you select an intercompany segment, you must complete at least one intercompany rule and check the Enable Intercompany Balancing option in the
Specify Ledger Options task for the Balancing API to perform intercompany balancing.

**Business Units Sheet**

Enter the name of your business unit.

You can enter more than one business unit per ledger but it is not recommended.

![Business Units Sheet](image)

**Note**

Enter a list of your legal entities. Include their registration number and assigned parent or child value.

**Companies and Legal Entities Sheet**

You can create up to 9 levels of parent values to use to roll up your legal entities to meet corporate and local reporting requirements.

![Companies and Legal Entities Sheet](image)

**Natural Accounts Sheet**

Enter your account values that are used to record the type of balance.

![Natural Accounts Sheet](image)

- **Parent and Child Values** with Descriptions are used to build hierarchies. Hierarchies are used for chart of accounts mappings, revaluations, data access sets, cross validation rules, and segment value security rules. The balances cube and account hierarchies are also used for financial reporting, Smart View queries, and allocations.

- **Account Type** is used to identify the type of account, Asset, Liability, Revenue, Expense, or Owner’s Equity. Account types are used in year end close processes and to correctly categorize your account balances for reporting.

- **Financial Category** (optional) is used to identify groups of accounts for reporting with Oracle Fusion Transactional Business Intelligence.
Upload the Sheets and Run Deployment

Return to the Chart of Accounts, Calendar, and Ledger sheet after completing the other sheets complete the following steps:

1. **(B) Generate Chart of Accounts File:** The program generates an XML data file for the entered chart of accounts and hierarchies setup data. Save the file to a network or local drive.

2. **(B) Generate Ledger, Legal Entity, and Business Units File:** The program generates an XML data file for the entered ledger, legal entities, and business unit setup data. Save the file a network or local drive.

3. **(N) Setup and Maintenance > Functional Setup Manager > Upload Chart of Accounts task.** The Upload Enterprise Structures process is launched.

4. **(B) Upload File.**

5. **(B) Browse.** Select the first file you saved: ChartOfAccounts.xml

6. **(B) Submit.**

7. Verify that the process was completed without errors or warnings.

8. **(N) Setup and Maintenance > Deploy Chart of Accounts task > (B) Deploy the Accounting Flexfield.**

9. **(I) Refresh** until the green check mark appears and verifies that the deployment is successful.

10. **(N) Setup and Maintenance > Upload Ledger, Legal Entities, and Business Units task.** The Upload Enterprise Structures process is launched.
11. **(B) Upload File.**

12. **(B) Browse.** Select the second file you saved: FinancialsCommonEntities.xml

13. **(B) Submit.**

14. Verify that the process was completed without errors or warnings.

---

**Tip**

You cannot change the chart of accounts, accounting calendar, or currency for your ledger after the setup is created. Assign the data role that was automatically generated for the ledger to your users. Then open the first period to begin entering data.

---

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

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

**Scenario**

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

**InFusion Corporation**

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

**Analysis**

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

**Chart of Accounts Model**

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

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

At the chart of accounts structure instance level, each segment is associated with a value set that conforms to the characteristic of that segment. For example, you assign a value set with the same segment type and length to each segment. You are using hierarchies with your chart of accounts segments. Each structure instance segment is assigned a tree code to indicate the source of the hierarchy information for the associated value set. The same value set can be used multiple times within the same or across different chart of accounts instances within the same structure or in different structures. This functionality reduces your segment value creation and maintenance across your charts of accounts.

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

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

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

Creating Chart of Accounts Structure and Instances: Examples

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

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

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

1. Navigate to the Manage Chart of Accounts page from the Functional Setup Manager by querying on Manage Chart of Accounts and clicking the Go To Task.

2. Select General Ledger from the Module list of values and click Search.

3. Click Manage Structures to open the Manage Key Flexfield Structures page.

4. Select the General Ledger row and click the Create to open the Create Key Flexfield Structure page.

5. Enter a unique Structure Code, INFUSION_AM_COA_STRUCTURE, and Name, InFusion America COA Structure. Provide an optional Description, InFusion America Inc. Chart of Accounts Structure.

6. Select the - Delimiter to visually separate your segment values.

7. Click Save.

8. To create a new segment, click the Create to open the Create Key Flexfield Segment page.

a. Enter the following parameters:

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

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

Note

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

c. Click Save and Close.

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

**Chart of Account Instance**

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

1. Navigate to the **Manage Chart of Accounts** page from the Functional Setup Manager by querying on **Manage Chart of Accounts** and clicking the **Go To Task**.

2. Select **General Ledger** from the Module list of values and click **Search**.

3. Select the **General Ledger** row and click **Manage Structure Instances** to open the **Manage Key Flexfield Structure Instance** page.

4. Click the **Create** icon to open the **Create Key Flexfield Structure Instance** page.

5. Enter a unique Structure Instance Code, **INFUSION_AM_COA_INSTANCE**, and Name, **InFusion America COA Instance**. Provide an optional Description, **InFusion America Inc. Chart of Accounts Structure Instance**.

6. Select **Dynamic combination creation allowed** to indicate that you want to dynamically generate account combinations.

7. Associate your instance with your Structure Name, **InFusion America Structure**.

---

**Note**

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

---

8. Click **Save**.

9. Optionally, select the segment row and click **Edit** to modify instance segments.

10. Check **Required**, **Displayed**, and **BI enabled** check boxes.

---

**Note**

Check the Required and Displayed options for all segments including those intended for future use. The recommended best practice is to define one segment for future use and set a default value. This ensures room for expansion in your chart of accounts and that the extra segment is populated in the account combinations.

Check the BI (Business Intelligence) enabled option to use key flexfield segments in Oracle Fusion Transactional Business Intelligence. The business intelligence check box is only valid when enabled on segments with segment labels. The second step is to populate the BI Object Name field for each of the segment labels in the Manage Segment Label page opened from the Manage Key Flexfields page.
11. Click OK.

12. Click **Save and Close**.

13. Define additional instances following the same process.

**Note**

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

14. Click **Done**.

15. Click **Deploy Flexfield**.

16. Click **OK**.

**Balancing Segments: Explained**

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

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

The following explains processes that use balancing segments.

- **Intercompany balancing**: Adds lines to unbalanced journals using intercompany rules.

- **Opening first period of the new accounting year**: Calculates retained earnings amounts at the level of granularity that totals revenue and expense account balances for multiple balancing segment value combinations. This applies to standard and average balances.

- **Importing journals**: Adds lines using the suspense account on unbalanced journals.

- **Posting journals**: Adds additional lines to unbalanced journals for the following enabled account types:
  - Suspense
  - Rounding
  - Net income
  - Retained earnings
• Cumulative translation adjustments from replication of revaluation journals to reporting currencies and for multiple reporting currency account type specific conversion

• Posting prior period journals: Calculates any income statement impact and posts to the appropriate retained earnings account.

• Translating balances: Supports multiple balancing segments for the following accounts:
  • Retained earnings: Calculated translated retained earnings are post to the retained earnings account by balancing segment. Retained earnings represents the summing of the translated revenue and expense accounts across multiple balancing segment values.
  • Cumulative translation adjustment: Amounts posted by balancing segment to these accounts represents currency fluctuation differences between ranges of accounts which use different rate types. For example, period end rates are used for asset and liability accounts and historical rates for equity accounts.

• Revaluing Balances: Supports multiple balancing segments when calculating gain or loss accounts.

• Creating Opening Balances: Initializes reporting currency balances by converting from the total primary currency. Any difference in the reporting currency amounts is offset by populating retained earnings accounts.

• Closing year end: Supports multiple balancing segments when calculating the income statement offset and closing account in the closing journals.

Multiple Balancing Segments: Points to Consider

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

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

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

Journal Entry Processing

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

**Implementation Timing**

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

**Note**

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

**Change Options**

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

**Migration Adjustments**

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

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

- Intercompany balancing
- Suspense posting
- Rounding imbalance adjustments on posting
- Entered currency balancing
- Revaluation gains or losses
- Retained earnings calculations at the opening of each new fiscal year
Define Enterprise Structures

- Cumulative translation adjustments during translation

Note

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

Creating One Chart of Accounts Structure with Many Instances: Example

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

Scenario

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

InFusion Corporation

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

Analysis

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

Chart of Accounts Model

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

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

At the chart of accounts structure instance level, each segment is associated with a value set that conforms to the characteristic of that segment. For example, you assign a value set with the same segment type and length to each segment. You are using hierarchies with your chart of accounts segments. Each structure instance segment is assigned a tree code to indicate the source of the hierarchy.
information for the associated value set. The same value set can be used multiple times within the same or across different chart of accounts instances within the same structure or in different structures. This functionality reduces your segment value creation and maintenance across your charts of accounts.

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

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

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

**Using Multiple Balancing Segments: Example**

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

**Scenario**

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

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

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

<table>
<thead>
<tr>
<th>Account</th>
<th>Debit</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company 1-Cost Center A-Advertising Expense Account</td>
<td>600</td>
<td></td>
</tr>
<tr>
<td>Company 2-Cost Center B-Advertising Expense Account</td>
<td></td>
<td>600</td>
</tr>
<tr>
<td>Company 1-Cost Center A-Phone Expense Account</td>
<td>800</td>
<td></td>
</tr>
</tbody>
</table>
During the posting process, the last four lines are created to balance the entry across the primary and second balancing segments, company and cost center.

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

**FAQs for Manage Chart of Accounts Structures and Structure Instances**

**How can I use future accounting segments?**

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

**Define Chart of Accounts for Enterprise Structures: Manage Chart of Accounts Value Sets and Value Set Values**

**Chart of Accounts Values Sets: Critical Choices**

A value set is the collection of account values that are associated with a segment of a chart of accounts structure instance. When creating values sets, consider the following critical choices:
• Module Designation
• Validation Type
• Format Assignments
• Security Rules
• Values Definition

Module Designation

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

Validation Type

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

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

Format Assignments

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

Security Rules

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

Value Definition

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

- Set the values to conform to the value set length and type.
• Enter the value, its description, and its attributes including the **Enable** check box, **Start Date**, and **End Date**.

• Assign the following attributes: **Parent** or **Summary** check box, **Posting is allowed**, and **Budgeting is allowed**.

**Note**

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

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

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

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

**Scenario**

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

1. Navigate to the **Manage Chart of Accounts Value Sets** task from within your implementation project and click the **Go to Task**.

2. Click the **Create** icon on the toolbar of the Search Results table. The **Create Value Set** page opens.

3. Enter a unique **Value Set Code**, **InFusion America Company**, and an **optional Description**, **Company values for InFusion America Inc.**

4. Select **General Ledger** from the list in the Module field.

5. Select **Independent** as Validation Type.

6. Select **Character** as the Validation Data Type.

7. Click **Save and Close**.

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

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

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

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

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

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

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

7. Click the Save button.

**Note**

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

- Dim - GL Segment1
Define Chart of Accounts of Enterprise Structures: Manage Accounting Calendars

Defining Accounting Calendars: Critical Choices

Define an accounting calendar to create your accounting year and the periods it contains. Specify common calendar options that the application uses to automatically generate a calendar with its periods. Specifying all the options makes defining a correct calendar easier and more intuitive with fewer errors. The choices you make when specifying the following options are critical, because it is difficult to change your accounting calendar after a period status is set to open or future enterable.

- Start Date
- Period Frequency
- Adjusting Period Frequency
- Period Name Format

Note

In Oracle Fusion, the common calendar types, monthly, weekly, 4-4-5, 4-5-4, 5-4-4, 4-week, quarterly, and yearly, are automatically generated. This functionality makes it easier to create and maintain accounting calendars. By using the period frequency option, you no longer have to go through the tedious task of defining each period manually.

Start Date

If you plan to run translation, specify a calendar start date that is a full year before the start date of the year of the first translation period for your ledger.
Translation cannot be run in the first period of a calendar. Consider how many years of history you are going to load from your previous system and back up the start date for those years plus one more. You cannot add previous years once the first calendar period has been opened.

**Period Frequency**

Use period frequency to set the interval for each subsequent period to occur, for example, monthly, quarterly, or yearly. If you select the period frequency of Other, by default, the application generates the period names, year, and quarter number. You specify the start and end dates. You must manually enter the period information. For example, select the period frequency of Other and enter 52 as the number of periods when you want to define a weekly calendar. For manually entered calendars, when you click the Add Year button, the application creates a blank year. Then, you must manually enter the periods for the new year. The online validation helps prevent erroneous entries.

If the year has been defined and validated, use the Add Year button to add the next year quickly. Accept or change the new rows as required. For example, with the Other frequency type calendar, dates may differ from what the application generates.

---

**Note**

In Oracle Fusion applications a calendar can only have one period frequency and period type. Therefore, if you have an existing calendar with more than one period type associated with it, during the upgrade from Oracle E-Business Suite, separate calendars are created based on each calendar name and period type combination.

---

**Adjusting Period Frequency**

Use the adjusting period frequency to control when the application creates adjusting periods. For example, some of the frequencies you select add one adjusting period at year end, two at year end, or one at the end of each quarter. The default is None which adds no adjusting periods. If you select the frequency of Other, the Number of Adjusting Periods field is displayed. Enter the number of desired adjusting periods and then, manually define them.

---

**Period Name Format Region**

The User-Defined Prefix field in the Period Name Format region is an optional feature that allows you to enter your own prefix. For example, define a weekly calendar and then enter a prefix of Week, - as the separator, and the period name format of Period numberYY fiscal year. The application creates the names of Week1-11, Week2-11, through Week52-11. The options for the Format field are predefined values. The list of values is filtered based on the selected separator and only displays the options that match the selected separator.

The year displayed in the period names is based on the selected period name format and the dates the period covers or if the period crosses years, on the year of the start date of the period. For example, April 10, 2010 to May 9, 2010 has the
period name of Apr-10 and December 10, 2010 to January 9, 2011 has the name of Dec-10. If period frequency is Other, then the period format region is hidden. The application generates a temporary period name for calendars with period frequency of Other, using a fixed format of Period numberYY. You can override this format with your own customized period names.

**Note**
For an accounting calendar that is associated with a ledger, changing period names or adding a year updates the accounting period dimension in the balances cubes.

---

**Calendar Validation: How It Works with the Accounting Calendar**

Calendar validation is automatic and prevents serious problems when you begin using the calendar. Once you set a calendar period status to open or future enterable, you cannot edit the period.

**Settings That Affect Calendar Validation**

The calendar validation runs automatically when you save the calendar.

**How the Calendar Is Validated**

The following table lists the validation checks performed when the accounting calendar is saved.

<table>
<thead>
<tr>
<th>Validation Performed</th>
<th>Example of Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique period number</td>
<td>2 assigned for two periods</td>
</tr>
<tr>
<td>Unique period name</td>
<td>Jan-11 entered twice</td>
</tr>
<tr>
<td>Period number beyond the maximum number of periods per year</td>
<td>13 for a 12 period calendar with no adjusting periods</td>
</tr>
<tr>
<td>Entered period name contains spaces</td>
<td>Jan 11</td>
</tr>
<tr>
<td>Single or double quotes in the period name</td>
<td>Jan '11</td>
</tr>
<tr>
<td>Nonadjusting periods with overlapping dates</td>
<td>01-Jan-2011 to 31-Jan-2011 and 30-Jan-2011 to 28-Feb-2011</td>
</tr>
<tr>
<td>Period date gaps</td>
<td>01-Jan-2011 to 28-Jan-2011 and 31-Jan-2011 to 28-Feb-2011</td>
</tr>
<tr>
<td>Missing period numbers</td>
<td>Periods 1 through 6 defined for a twelve month calendar</td>
</tr>
<tr>
<td>Period number gaps</td>
<td>1, 3, 5</td>
</tr>
<tr>
<td>Period numbers not in sequential order by date</td>
<td>Period 1 covers 01-Jan-2011 to 31-Jan-2011 and period 2 covers 01-Mar-2011 to 31-Mar-2011, and period 3 covers 01-Feb-2011 to 28-Feb-2011.</td>
</tr>
<tr>
<td>Quarter number gaps</td>
<td>1, 3, 4</td>
</tr>
<tr>
<td>Quarters not in sequential order by period</td>
<td>1, 3, 2, 4</td>
</tr>
<tr>
<td>Period start or end dates more than one year before or after the fiscal year</td>
<td>July 1, 2010 in a 2012 calendar</td>
</tr>
</tbody>
</table>
FAQs for Manage Accounting Calendars

How can I identify errors in my accounting calendar?

Oracle Fusion General Ledger identifies erroneous entries online as you enter a new calendar or change data on an existing calendar. The application also automatically validates the data when you save the calendar.

What’s the difference between calendar and fiscal period naming?

The period naming format determines the year that is appended to the prefix for each period in the calendar. For the example, your accounting year has a set of twelve accounting period with a start date of September 1, 2011 and the end date is August 31, 2012, with each period’s date range following the natural calendar month date range.

Calendar period naming format: Select the calendar period format to append the period’s start date’s year to the prefix. For the period covering September 1, 2011 to December 31, 2011, then 2011 or just 11, depending on the period format selected, is appended to each period’s name. For the remaining periods covering January 1, 2012 to August 31, 2012, then 2012 or 12, is appended to each period’s name.

Fiscal period naming format: Select the fiscal period format to always append the period’s year assignment to the prefix. If the accounting periods in the set of twelve are all assigned the year of 2012, then 2012 or just 12, depending on the period format selected, is appended to the period name of all 12 periods.

When do I update an existing calendar?

Update an existing calendar before the new periods are needed as future periods, based on the future period setting in your accounting configuration. If a complete year has been defined and validated, use the Add Year button to add the next year quickly. Accept or change the new rows as required. For example, with the Other frequency type calendar, dates may differ from what the application generates.

What happens if I upgrade my calendar from Oracle E-Business Suite Release 12?

The migration script assigns a period frequency that most closely matches your Oracle E-Business Suite Release 12 calendar. When you use the Oracle Fusion applications Add Year functionality for the first time, you have an opportunity to review and change the period frequency. The Calendar Options page opens only for calendars upgraded from Release 12 to allow one time modification.

Make your changes to the period frequency, adjusting period frequency, and period name format, including the prefix and separator, as needed. Changes can not conflict with the existing upgraded calendar definition. Update the calendar
name and description in the calendar header, as needed, for all calendars. Period details for a new year will be generated automatically based on the latest calendar options. You can also manually update the calendar. The modified calendar options affect future years only.

Define Accounting Configurations of Enterprise Structures: Manage Primary or Secondary Ledgers

Accounting Configuration Offerings: Overview

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

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

There are two offerings predefined for financial implementations.

- The Oracle Fusion Accounting Hub offering is used to add the Oracle Fusion General Ledger and Oracle Fusion Subledger Accounting application features to an existing enterprise resource planning (ERP) system to enhance the current reporting and analysis.
- The Oracle Fusion Financials offering, which includes the Oracle Fusion General Ledger and Oracle Fusion Subledger Accounting application features, as well as at least one of the subledger financial applications.

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

Ledgers and Subledgers: Explained

Oracle Fusion Applications reflect the traditional segregation between the general ledger and associated subledgers. Detailed transactional information is captured in the subledgers and periodically imported and posted in summary or detail to the ledger.

A ledger determines the currency, chart of accounts, accounting calendar, ledger processing options, and accounting method for its associated subledgers. Each accounting setup requires a primary ledger and optionally, one or more secondary ledgers and reporting currencies. Reporting currencies are associated with either a primary of secondary ledger.
The number of ledgers and subledgers is unlimited and determined by your business structure and reporting requirements.

**Single Ledger**

If your subsidiaries all share the same ledger with the parent company or they share the same chart of accounts and calendar, and all reside on the same applications instance, you can consolidate financial results in Oracle Fusion General Ledger in a single ledger. Use Oracle Fusion Financial Reporting functionality to produce individual entity reports by balancing segments. General Ledger has three balancing segments that can be combined to provide detailed reporting for each legal entity and then rolled up to provide consolidated financial statements.

**Multiple Ledgers**

Accounting operations using multiple ledgers can include single or multiple applications instances. You need multiple ledgers if one of the following is true:

- You have companies that require different account structures to record information about transactions and balances. For example, one company may require a six-segment account, while another needs only a three-segment account structure.
- You have companies that use different accounting calendars. For example, although companies may share fiscal year calendars, your retail operations require a weekly calendar, and a monthly calendar is required for your corporate headquarters.
- You have companies that require different functional currencies. Consider the business activities and reporting requirements of each company. If you must present financial statements in another country and currency, consider the accounting principles to which you must adhere.

**Subledgers**

Oracle Fusion Subledgers capture detailed transactional information, such as supplier invoices, customer payments, and asset acquisitions. Oracle Fusion Subledger Accounting is an open and flexible application that defines the accounting rules, generates detailed journal entries for these subledger transactions, and posts these entries to the general ledger with flexible summarization options to provide a clear audit trail.

**Ledgers: Points to Consider**

Companies account for themselves in primary ledgers, and, if necessary, secondary ledgers and reporting currencies. Your transactions from your subledgers are posted to your primary ledgers and possibly, secondary ledgers or reporting currencies. Local and corporate compliance can be achieved through an optional secondary ledger, providing an alternate accounting method, or in some cases, a different chart of accounts. Your subsidiary’s primary and
secondary ledgers can both be maintained in your local currency, and you can convert your local currency to your parent’s ledger currency to report your consolidated financial results using reporting currencies or translation.

**Primary Ledgers**

A primary ledger is the main record-keeping ledger. Like any other ledger, a primary ledger records transactional balances by using a chart of accounts with a consistent calendar and currency, and accounting rules implemented in an accounting method. The primary ledger is closely associated with the subledger transactions and provides context and accounting for them.

To determine the number of primary ledgers, your enterprise structure analysis must begin with your financial, legal, and management reporting requirements. For example, if your company has separate subsidiaries in several countries worldwide, enable reporting for each country’s legal authorities by creating multiple primary ledgers that represent each country with the local currency, chart of accounts, calendar, and accounting method. Use reporting currencies linked to your country specific primary ledgers to report to your parent company from your foreign subsidiaries. Other considerations, such as corporate year end, ownership percentages, and local government regulations and taxation, also affect the number of primary ledgers required.

**Secondary Ledgers**

A secondary ledger is an optional ledger linked to a primary ledger for the purpose of tracking alternative accounting. A secondary ledger can differ from its primary ledger by using a different accounting method, chart of accounts, accounting calendar, currency, or processing options. All or some of the journal entries processed in the primary ledger are transferred to the secondary ledger, based on your configuration options. The transfers are completed based on the conversion level selected. There are four conversion levels:

- **Balance**: Only Oracle Fusion General Ledger balances are transferred to the secondary ledger.
- **Journal**: General Ledger journal posting process transfers the journal entries to the secondary ledger.
- **Subledger**: Oracle Fusion Subledger Accounting creates subledger journals to subledger level secondary ledgers as well as reporting currencies.
- **Adjustments Only**: Incomplete accounting representation that only holds adjustments. The adjustments can be manual or detailed adjustments from Subledger Accounting. This type of ledger must share the same chart of accounts, accounting calendar, and period type combination, and currency as the associated primary ledger.

**Note**

A full accounting representation of your primary ledger is maintained in any subledger level secondary ledger.

Secondary ledgers provide functional benefits, but produce large volumes of additional journal entry and balance data, resulting in additional performance
and memory costs. When adding a secondary ledger, consider your needs for secondary ledgers or reporting currencies, and select the least costly data conversion level that meets your requirements. For secondary ledgers, the least costly level is the adjustment data conversion level because it produces the smallest amount of additional data. The balance data conversion level is also relatively inexpensive, depending upon how often the balances are transferred from the primary to the secondary ledger. The journal and subledger data conversion levels are much more expensive, requiring duplication of most general ledger and subledger journal entries, as well as general ledger balances.

For example, you maintain a secondary ledger for your International Financial Reporting Standards (IFRS) accounting requirements, while your primary ledger uses US Generally Accepted Accounting Principles (GAAP). You decided to select the subledger level for your IFRS secondary ledger. However, since most of the accounting is identical between US GAAP and IFRS, a better solution is to use the adjustment only level for your secondary ledger. The subledger level secondary ledger requires duplication of most subledger journal entries, general ledger journal entries, and general ledger balances. With the adjustment only level, your secondary ledger contains only the adjustment journal entries and balances necessary to convert your US GAAP accounting to the IFRS accounting, which uses a fraction of the resources that are required by full subledger level secondary ledger.

Following are scenarios that may require different combinations of primary and secondary ledgers:

- The primary and secondary ledgers use different charts of accounts to meet varying accounting standards or methods. A chart of accounts mapping is required to instruct the application how to propagate balances from the source (primary) chart of accounts to the target (secondary) chart of accounts.

- The primary and secondary ledgers use different accounting calendars to comply with separate industry and corporate standards.

**Note**

Use the same currency for primary and secondary ledgers to avoid difficult reconciliations, if you have the resources to support the extra posting time and data storage. Use reporting currencies or translations to generate the different currency views needed to comply with internal reporting needs and consolidations.

**Reporting Currencies**

Reporting currencies maintain and report accounting transactions in additional currencies. Each primary and secondary ledger is defined with a ledger currency that is used to record your business transactions and accounting data for that ledger. It is advisable to maintain the ledger in the currency in which the majority of its transactions are denominated. For example, create, record, and close a transaction in the same currency to save processing and reconciliation time. Compliance, such as paying local transaction taxes, is also easier using a local currency. Many countries require that your accounting records be kept in their national currency.
If you need to maintain and report accounting records in different currencies, you do this by defining one or more reporting currencies for the ledger. There are three conversion levels for reporting currencies:

- **Balance**: Only General Ledger balances are converted into the reporting currency using translation.
- **Journal**: General Ledger journal entries are converted to the reporting currency during posting.
- **Subledger**: Subledger Accounting creates subledger reporting currency journals along with primary ledger journals.

**Note**

A full accounting representation of your primary ledger is maintained in any subledger level reporting currency. Secondary ledgers cannot use subledger level reporting currencies.

Of the three data conversion levels available, the balance data conversion level is typically the least expensive, requiring duplication of only the balance level information. The journal and subledger data conversion levels are more expensive, requiring duplication of most general ledger and subledger journal entries, as well as general ledger balances.

Do not use journal or subledger level reporting currencies if your organization has only an infrequent need to translate your financial statements to your parent company’s currency for consolidation purposes. Standard translation functionality meets this need. Consider using journal or subledger level reporting currencies when any of the following conditions exist.

- You operate in a country whose unstable currency makes it unsuitable for managing your business. As a consequence, you need to manage your business in a more stable currency while retaining the ability to report in the unstable local currency.
- You operate in a country that is part of the European Economic and Monetary Union (EMU), and you choose to account and report in both the European Union currency and your National Currency Unit (NCU).

**Note**

The second option is rare since most companies have moved beyond the initial conversion to the EMU currency. However, future decisions could add other countries to the EMU, and then, this option would again be used during the conversion stage.

**Financial Ledgers: How They Fit Together**

Oracle Fusion Applications is an integrated suite of business applications that connects and automates the entire flow of the business process across both front and back office operations and addresses the needs of a global enterprise. The process of designing the enterprise structure, including the accounting configuration, is the starting point for an implementation. This process often includes determining financial, legal, and management reporting requirements, setting up primary and secondary ledgers, making currency choices, and examining consolidation considerations.
This figure shows the enterprise structure components and their relationships to each other. Primary ledgers are connected to reporting currencies and secondary ledgers to provide complete reporting options. Legal entities are assigned to ledgers, both primary and secondary, and balancing segments are assigned to legal entities. Business units must be connected to both a primary ledger and a default legal entity. Business units can record transactions across legal entities.

**Primary Ledgers**

A primary ledger is the main record-keeping ledger. Create a primary ledger by combining a chart of accounts, accounting calendar, ledger currency, and accounting method. To determine the number of primary ledgers, your enterprise structure analysis must begin with determining financial, legal, and management reporting requirements. For example, if your company has separate subsidiaries in several countries worldwide, create multiple primary ledgers representing each country with the local currency, chart of accounts, calendar, and accounting method to enable reporting to each country's legal authorities.

If your company just has sales in different countries, with all results being managed by the corporate headquarters, create one primary ledger with multiple balancing segment values to represent each legal entity. Use secondary ledgers or reporting currencies to meet your local reporting requirements, as needed. Limiting the number of primary ledgers simplifies reporting because consolidation is not required. Other considerations such as corporate year end, ownership considerations, and local government regulations, also affect the number of primary ledgers required.

**Secondary Ledgers**

A secondary ledger is an optional ledger linked to a primary ledger. A secondary ledger can differ from its related primary ledger in chart of accounts, accounting
calendar, currency, accounting method, or ledger processing options. Reporting requirements, for example, that require a different accounting representation to comply with international or country-specific regulations, create the need for a secondary ledger.

Below are scenarios and required action for different components in primary and secondary ledgers:

- If the primary and secondary ledgers use different charts of accounts, the chart of accounts mapping is required to instruct the system how to propagate journals from the source chart of accounts to the target chart of accounts.

- If the primary and secondary ledgers use different accounting calendars, the accounting date and the general ledger date mapping table will be used to determine the corresponding non-adjusting period in the secondary ledger. The date mapping table also provides the correlation between dates and non-adjusting periods for each accounting calendar.

- If the primary ledger and secondary ledger use different ledger currencies, currency conversion rules are required to instruct the system on how to convert the transactions, journals, or balances from the source representation to the secondary ledger.

Note: Journal conversion rules, based on the journal source and category, are required to provide instructions on how to propagate journals and types of journals from the source ledger to the secondary ledger.

**Reporting Currencies**

Reporting currencies are the currency you use for financial, legal, and management reporting. If your reporting currency is not the same as your ledger currency, you can use the foreign currency translation process or reporting currencies functionality to convert your ledger account balances in your reporting currency. Currency conversion rules are required to instruct the system on how to convert the transactions, journals, or balances from the source representation to the reporting currency.

**Legal Entities**

Legal entities are discrete business units characterized by the legal environment in which they operate. The legal environment dictates how the legal entity should perform its financial, legal, and management reporting. Legal entities generally have the right to own property and the obligation to comply with labor laws for their country. They also have the responsibility to account for themselves and present financial statements and reports to company regulators, taxation authorities, and other stakeholders according to rules specified in the relevant legislation and applicable accounting standards. During setup, legal entities are assigned to the accounting configuration, which includes all ledgers, primary and secondary.

**Balancing Segments**

You assign primary balancing segment values to all legal entities before assigning values to the ledger. Then, assign specific primary balancing segment values to the primary and secondary ledgers to represent nonlegal entity related
transactions such as adjustments. You can assign any primary balancing segment value that has not already been assigned to a legal entity. You are allowed to assign the same primary balancing segment values to more than one ledger. The assignment of primary balancing segment values to legal entities and ledgers is performed within the context of a single accounting setup. The Balancing Segment Value Assignments report is available to show all primary balancing segment values assigned to legal entities and ledgers across accounting setups to ensure the completeness and accuracy of their assignments. This report allows you to quickly identify these errors and view any unassigned values.

Business Units

A business unit is a unit of an enterprise that performs one or many business functions that can be rolled up in a management hierarchy. When a business function produces financial transactions, a business unit must be assigned a primary ledger, and a default legal entity. Each business unit can post transactions to a single primary ledger, but it can process transactions for many legal entities. Normally, it will have a manager, strategic objectives, a level of autonomy, and responsibility for its profit and loss. You define business units as separate task generally done after the accounting setups steps.

The business unit model:

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

For example, if your company requires business unit managers to be responsible for managing all aspects of their part of the business, then consider using two balancing segments, company and business unit to enable the production of business unit level balance sheets and income statements.

Transactions are exclusive to business units. In other words, you can use business unit as a securing mechanism for transactions. For example, if you have an export business that you run differently from your domestic business, use business units to secure members of the export business from seeing the transactions of the domestic business.

Creating Primary Ledgers: Example

Create a primary ledger as your main record-keeping ledger. Like any other ledger, a primary ledger records transactional balances by using a chart of accounts with a calendar, currency, and accounting rules implemented in an accounting method. The primary ledger is closely associated with the subledger transactions and provides context and accounting for them.

Scenario

Your company, InFusion Corporation is implementing Oracle Fusion Applications. You have been assigned the task of creating a primary ledger for your InFusion America entity.
1. Navigate to the **Define Accounting Configurations** task list and open Manage Primary Ledgers from within your implementation project. Click the **Go to Task**.

2. Click the **Create** icon.

3. Enter the following values:

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

4. Click **Save and Edit Task List** to navigate back to the accounting configuration task list.

**Note**

You cannot change the chart of accounts, accounting calendar, or currency for your ledger after you save your ledger.

---

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

**Specifying Ledger Options: Worked Example**

This example demonstrates specifying the ledger options for your primary ledger. Your company, InFusion Corporation, is a multinational conglomerate that operates in the United States (US) and the United Kingdom (UK). InFusion has purchased an Oracle Fusion enterprise resource planning (ERP) solution including Oracle Fusion General Ledger and all of the Oracle Fusion subledgers.

After completing your InFusion America Primary Ledger, select **Specify Ledger Options** under the Define Accounting Configuration task list on the Functional Setup Manager page.

**Note**

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

**Reviewing General Region Options**

1. Accept the **Name** and **Description** defaults for the ledger selected.
2. Review the Currency and Chart of Accounts for the specified ledger, which are automatically populated.

Setting Accounting Calendar Region Options
1. Review the Accounting Calendar that defaults from your ledger.
2. Select Jan-2011 as the First Open Period for your ledger.
   Important: Select a period after the first defined period in the ledger calendar to enable running translation. You cannot run translation in the first defined period of a ledger calendar. In this example, your calendar began with Jan-2010.
3. Enter 3 for the Number of Future Enterable Periods.
   Any value between 0 and 999 periods can be specified to permit entering journals but not posting them in future periods. Minimize the number of open and future periods to prevent entry in the wrong period.

Selecting the Subledger Accounting Region Options
1. Accept the default Accounting Method from your ledger.

Completing the Period Close Region Options
1. Enter your Retained Earnings Account: 101-00-31330000-0000-000-0000-0000.
   This account is required for the General Ledger to perform the movement of revenue and expense account balances to this account at the end of the accounting year.
2. Enter your Cumulative Translation Adjustment Account: 101-00-31350000-0000-000-0000-0000.
   Note: The Cumulative Translation Adjustment (CTA) account is required for ledgers running translation.
3. Do not enter a Default Period End Rate Type or Default Period Average Rate Type.
   The values entered here are used as the default for balance level reporting currency processing. InFusion America Primary Ledger is using the subledger level reporting currency processing.

Specifying the Journal Processing Region Options
1. Specify the Balance options as outlined in the following table.

<table>
<thead>
<tr>
<th>Option</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Suspense</td>
<td>General Ledger</td>
</tr>
<tr>
<td>Default Expense Account</td>
<td>101-00-98199999-0000-000-0000-0000</td>
</tr>
<tr>
<td>Rounding Account</td>
<td>101-10-98189999-0000-000-0000-0000</td>
</tr>
<tr>
<td>Entered Currency Balancing Account</td>
<td>101-10-98179999-0000-000-0000-0000</td>
</tr>
<tr>
<td>Balancing Threshold Percent</td>
<td>10</td>
</tr>
</tbody>
</table>
2. Click all the following Entry options listed in the table.

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

3. Click the **Separate journals by accounting date during journal import** for the Import option to create individual journal entries for each accounting date.

4. For the Reversal options, select InFusion America Accrual Set from the list of values in the **Journal Reversal Criteria Set** field and click the **Launch AutoReverse after open period** to reverse accrual journal entries automatically when a new period is opened.

5. Click the **Enable intercompany accounting** for the Intercompany option to enable automatic balancing by the application for primary, second, and third balancing segments (if implemented) on intercompany journal entries and transactions.

   Note: To complete the intercompany accounting functionality, you must define intercompany rules.

### Assigning Legal Entities and Balancing Segments: Examples

Optionally, assign legal entities and balancing segments to your accounting configuration.

#### Assign Legal Entities

Assign one or more legal entities to your configuration by following these steps:

1. Navigate to the **Assign Legal Entities** task. Click the **Go to Task**.
2. Click the **Select and Add** icon.
3. Click **Search**. Select your legal entities.
4. Click **Apply**. Click **Done**.
5. Click **Save and Close**.

#### Assign Balancing Segments to Legal Entities

Assign balancing segment values to your legal entities by following these steps:
1. Navigate to the Assign Balancing Segment Values to Legal Entities task. Click the Go to Task.
2. Click the Create icon.
3. Select the balancing segment value. Optionally, add a Start Date.
4. Click Save and Close to close the create page.
5. Click Save and Close.

Assign Balancing Segments to Ledgers

Assign balancing segment values directly to your ledger by following these steps:

1. Navigate to the Assign Balancing Segment Value to Ledger task. Click the Go to Task.
2. Select the balancing segment value.
3. Optionally enter a start date.
4. Click Save and Close.

Note

The balancing segment values that are assigned to the ledger represent nonlegal entity transactions, such as adjustments. If you use legal entities, you must assign balancing segment values to all legal entities before assigning values to the ledger. The only available balancing segment values that can be assigned to ledgers are those not assigned to legal entities.

Define Accounting Configurations of Enterprise Structures: Manage Reporting Currencies

Reporting Currency Balances: How They Are Calculated

Reporting currency balances, set at the journal or subledger level, are updated when journal entries that originate in Oracle Fusion General Ledger are posted and converted to your reporting currencies. This process includes General Ledger manual journals, periodic journals, and allocations, and at the subledger level, journals from Oracle Fusion Subledger Accounting and imported from sources other than your Oracle Fusion subledgers. When you post a journal in a ledger that has one or more reporting currencies defined, the posting process creates new journals converted to each of your reporting currencies and includes them in the same batch as the original journal with a status of Posted.

Settings That Affect

Reporting currencies share a majority of the ledger options with their source ledger. For example, the reporting currency uses the same suspense account
Define Enterprise Structures

and retained earnings accounts as its source ledger. However, there are certain options that need to be set specifically for the reporting currencies. For example, reporting currencies are maintained at one of three currency conversion levels:

- **Balance Level**: Only balances are maintained in the reporting currency using the General Ledger Translation process.

- **Journal Level**: Journal entries and balances are converted to the reporting currency by the General Ledger Posting process.

- **Subledger Level**: Subledger Accounting creates reporting currency journals for subledger transactions. General Ledger converts journals that originated in General Ledger or that are imported from sources other than the Oracle Fusion subledgers. The full accounting representation of your primary ledger is maintained in the subledger level reporting currency.

---

**Note**

Secondary Ledgers cannot use subledger level reporting currencies.

There are multiple dependencies between a reporting currency and its source ledger. Therefore, it is important that you complete your period opening tasks, daily journal or subledger level reporting currencies accounting tasks, and period closing tasks in the correct order. Some guidelines are presented in the table below.

<table>
<thead>
<tr>
<th>Type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Period Opening Tasks</strong></td>
<td>Open the accounting period in both your ledger and reporting currencies before you create or import journals for the period. Converted journals are only generated in your reporting currency if the period is open or future enterable.</td>
</tr>
<tr>
<td><strong>Daily Tasks</strong></td>
<td>Enter the daily conversion rates to convert your journals to each of your reporting currencies.</td>
</tr>
</tbody>
</table>
| **Period Closing Tasks** | • Finish entering all regular and adjusting journals for the period in your ledger.  
                              • Post all unposted journals in your ledger if not already done in the previous step.  
                              • Post all unposted journals in your reporting currencies if not already done in the previous step.  
                              • Run Revaluation in both your ledger and reporting currencies. Post the resulting revaluation batches in each ledger.  
                              • As needed, translate balances in your ledger.  
                              • Generate needed reports from both your ledger and reporting currencies.  
                              • Close your accounting period in both your ledger and reporting currencies. |
How Reporting Currencies Are Calculated

If you use reporting currencies at the journal or subledger level, when you create accounting, post journal entries, or translate balances, journals are posted in your reporting currency. General Ledger and Subledger Accounting automatically generate journals in your reporting currencies where the entered currency amounts are converted to the reporting currency amounts. Other factors used in the calculation of reporting currency balances are listed:

- **Manual Journals**: Enter a manual journal batch in your reporting currency at the journal or subledger level by using the Create Journals page. Select the journal or subledger level reporting currency from the ledger’s list of values and continue in the same manner as entering any other manual journal.

- **Conversion Rounding**: Use the reporting currency functionality to round converted and accounted amounts using the same rounding rules used throughout your Oracle Fusion Applications. The reporting currency functionality considers several factors that are a part of the currencies predefined in your applications, including:
  - **Currency Precision**: Number of digits to the right of the decimal point used in currency transactions.
  - **Minimum Accountable Unit**: Smallest denomination used in the currency. This might not correspond to the precision.

- **Converted Journals**: Generate and post automatically, using the General Ledger Posting process, journals in your reporting currencies when you post the original journals in the source ledger for the following types of journals:
  - Manual journals
  - Periodic and allocation journals
  - Unposted journals from non-Oracle subledger applications
  - Unposted journals from any Oracle Fusion subledger that does not support reporting currency transfer and import
  - Optionally, revaluation journals

- **Unconverted Journals**: Rely on the subledger accounting functionality to converted and transfer Oracle Fusion subledger journals for both the original journal and the reporting currency journal to the General Ledger for import and posting. The reporting currency conversion for these journals is not performed by the General Ledger.

- **Approving Journals**: Use the journal approval feature to process reporting currency journals through your organization's approval hierarchy. You can enable journal approval functionality separately in your source ledger and reporting currencies.

- **Document Numbers**: Accept the default document numbers assigned by the General Ledger application to your journal when you enter a journal in your ledger. The converted journal in the reporting currency is
assigned the same document number. However, if you enter a journal in the reporting currency, the document number assigned to the journal is determined by the reporting currency.

- Sequential Numbering: Enable sequential numbering if you want to maintain the same numbering in your reporting currency and source ledger for journals, other than those journals for Oracle Fusion subledgers. Do not create separate sequences for your reporting currencies. If you do, the sequence defined for the reporting currencies is used and can cause document numbers not to be synchronized between the ledger and reporting currencies.

**Note**

If the Sequential Numbering profile option is set to **Always Used** or **Partially Used** and you define an automatic document numbering sequence, General Ledger enters a document number automatically when you save your journal. If you use manual numbering, you can enter a unique document number.

- Revaluation: Run periodically revaluation in your ledger and reporting currencies as necessary to satisfy the accounting regulations of the country in which your organization operates.

- Account Inquiries: Perform inquiries in the reporting currency. Drill down to the journal detail that comprises the reporting currency balance. If the journal detail is a converted journal that was converted automatically when the original journal was posted in the source ledger, you can drill down further to see the source ledger currency journal amounts.

**Note**

Be careful when changing amounts in a reporting currency, since the changes are not reflected in your source ledger. Making journal entry changes to a reporting currency makes it more difficult to reconcile your reporting currency to your source ledger. In general, enter or change your journals in your source ledger, and then allow posting to update the reporting currency.

**Note**

If you use reporting currencies at the journal or subledger level, statistical journals are generated for your reporting currencies, but the journals are not affected by the currency conversion process.

---

**Define Business Units: Manage Business Units**

**Business Units: Explained**

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

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

The Oracle Fusion Applications business unit model:

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

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

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

In summary, use business units in the following ways:

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

**Brief Overview of Business Unit Security**

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

For example, if a payables invoicing business function is enabled, then it is clear that there are employees in this business unit that perform the function
Define Business Units: Assign Business Unit Business Function

Business Functions: Explained

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

Business Functions

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

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

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

Note

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

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

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

When a business function produces financial transactions, a business unit must be assigned to a primary ledger, and a default legal entity. Each business unit can post transactions to a single primary ledger, but it can process transactions for many legal entities.

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

• Billing and revenue management
• Collections management
• Customer payments
• Expense management
• Materials management
• Payables invoicing
• Project accounting
• Receiving
• Requisitioning
**Business Unit Hierarchy: Example**

For example, your InFusion America Company provides:

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

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

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

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

**Define Business Units: Manage Service Provider Relationships**

**Shared Service Centers: Explained**

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

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

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

Define Business Units: Specify Customer Contract Management Business Function Properties

Customer Contracts Business Unit Setup: Explained

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

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

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

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

Enabling Related Customer Accounts

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

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

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

In the Invoice Currency to Ledger Currency region:

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

Managing Project Billing Options

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

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

Set these options, which apply to all contracts:

- Select the Transfer Revenue to General Ledger option if you want to create revenue accounting events and entries, and transfer revenue journals to the general ledger. If this option is not selected, then revenue can still be generated, but will not be transferred to the general ledger.
- Indicate if a reason is required for credit memos that are applied to invoices.

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

- Select an invoice numbering method, either Manual or Automatic. The invoice numbering method is the method that Oracle Fusion Receivables uses to number its invoices, upon release of draft invoices from Project Billing.
- If the invoice numbering method is Manual, then select an invoice number type, which sets the type of Receivables invoice numbers that are allowed. Valid values are Alphanumeric and Numeric.
- If the invoice numbering method is Automatic, then enter the next invoice number to use when generating Receivables invoice numbers.
- Select the Receivables batch source to use when transferring invoices to Receivables.
Set this option only for customer billing:

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

Managing Clause Numbering

You can choose to number clauses manually or automatically.

If you choose the automatic numbering method, you must select a determinant
level for the numbering. You must then select the appropriate clause sequence
category from document sequences that you set up for this numbering level.

Define Business Units: Specify Supplier Contract Management
Business Function Properties

Supplier Contracts Business Unit Setup: Explained

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

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

Managing Contract Terms Library Setup Options

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

Define Facilities: Manage Facility Shifts, Workday Patterns, and
Schedules

Schedule Components: How They Fit Together

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

Begin by creating shifts and then assigning those shifts to workday patterns.
Next, create a schedule that is a collection of workday patterns and any
exception dates.
Shift
A shift is a period of time, typically expressed in hours, and it can be defined by a start time and an end time, or a duration. A shift can be for a work period or an off period. You can create time, duration, and elapsed shifts.

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

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

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

Define Facilities: Manage Inventory Organizations

Inventory Organizations: Explained

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

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

- Item management
- Item and inventory management

Item Management

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

Item and Inventory Management

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

Define Facilities: Manage Item Organizations

Item Organization: Explained

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

**Note**

- Items belong to an item organization.
- Item attributes that are associated with financial and accounting information are hidden from the item if it exists within the item organization.
- Item organizations can be changed by administrators to an inventory organization by updating the necessary attributes. There is no difference in the way items are treated in these two types of organizations except that there cannot be any financial transactions in the downstream applications for items that are assigned to an item organization.
Define Social Networking

How can I enable social networking features?

Set the Social Networking Features Enabled profile option to "Y" on the Manage Social Networking Profile Option Values page. If you enable this profile option, users can access the following features:

- Kudos
- Message board
- Activity stream
- Connections
- Personal status

Note

If you disable social networking features, users can continue to access the bookmarks they created from the Bookmarks region in the Activities and Interests card on the My Portrait tab. Also, the spaces that users are a member of continue to appear in the Spaces region.
Define Security for Supply Chain Management

Security Tasks: Highlights

Security tasks include the following:
- Security setup
- Security implementation and administration

Note
Security setup and administration tasks typically use integrated user interface pages that are provided by the following products:
- Oracle Identity Manager (OIM)
- Oracle Authorization Policy Manager (APM)
- Oracle Fusion Human Capital Management (HCM) products
- Oracle Application Access Control Governor (AACG) in Oracle Enterprise Governance, Risk and Compliance (GRC)

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. See: Creating Users and Groups
- 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. For environments that are not in Oracle Cloud, use the super user account that was created during installation to sign in for the first time.

- Installation establishes the super user account. Refer to the Oracle Fusion Applications Installation Guide.
  
  See: Identity Management Configuration

- Oracle provides an initial user for accessing your services in Oracle Cloud. For more information, refer to "Oracle Cloud Application Services Security: Explained" in Oracle Cloud documentation.

- Synchronize LDAP users with HCM user management by performing the Run User and Roles Synchronization Process task. Monitor completion of the predefined Enterprise Scheduler process called Retrieve Latest LDAP Changes.

- Refer to information about creating person records in Oracle Fusion Applications Workforce Development Implementation Guide, or refer to the Oracle Fusion Middleware User's Guide for Oracle Identity Manager.
  
  See: Managing Users

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

**Required Security Administration Tasks**

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

- Perform the initial security tasks. If entitled to do so, see Initial Security Administration: Critical Choices.
  
  - Sign in to Oracle Fusion Applications using the IT security manager's or administrator's user name and password, and create and provision users who manage your implementation projects and set up enterprise structures by performing the Create Implementation Users task. Refer to the Oracle Fusion Middleware User's Guide for Oracle Identity Manager.

  See: User Management Tasks

- Create a data role for implementation users who will set up HCM that grants access to data in secured objects required for performing HCM setup steps. Provision the implementation user with this View All data role. See "Creating an HCM Data Role: Worked Example."

- For an overview of security tasks from the perspective of an applications administrator, refer to the Oracle Fusion Applications Administrator's Guide.

  See: Securing Oracle Fusion Applications
Optional Security Administration Tasks

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

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

  See: Creating Users

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

  See: Adding and Removing Roles

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

  See: User Interfaces


  See: Managing Oracle Fusion Applications Data Security Policies

- You manage role provisioning rules in Human Capital Management (HCM). Refer to the Role Mappings: Explained topic in the Oracle Fusion Applications Workforce Development Implementation Guide.


- 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 Oracle Enterprise Governance, Risk and Compliance (GRC), 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
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.

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 Enterprise Governance, Risk and Compliance (GRC) 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 Oracle Sales Cloud 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 Initiate HCM Spreadsheet Load task. If user and role provisioning rules have been defined, the Initiate HCM Spreadsheet Load process automatically creates user and role provisioning requests as the workers are created. Once the enterprise is set up, performing the Initiate HCM Spreadsheet Load 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 Initiate HCM Spreadsheet Load task results in new user accounts being created. Worker email addresses as an alternate input for the Initiate HCM Spreadsheet Load 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 Initiate HCM Spreadsheet Load 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 Oracle Sales Cloud.

- 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 Initiate HCM Spreadsheet Load 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, Oracle Sales Cloud, 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
Define Security for Supply Chain Management

- 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 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 following enterprise roles are provisioned to a single super user that is set up by the Oracle Fusion Applications installation process, and to the initial user set up by Oracle for Oracle Cloud Application Services:
  - Application Implementation Consultant
  - IT Security Manager
  - Application Administrators for the provisioned products

Initial security administration also includes provisioning the IT Security Manager role with Oracle Identity Management (OIM) roles for user and role management.
  - Identity User Administrator
  - Role Administrator

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

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

Tools Used to Perform Security Tasks
Security tasks are supported by tools within both Oracle Fusion Applications and Oracle Fusion Middleware.
The figure lists the tasks associated with each of the integrated products and pillars of an Oracle Fusion Applications deployment.
Security Tasks: Overview

Security tasks span multiple business processes and are performed by various roles using numerous integrated tools.

The following table shows the business process model (BPM) tasks and tools used to support securing Oracle Fusion Applications.

<table>
<thead>
<tr>
<th>Example Task</th>
<th>Oracle BPM Task</th>
<th>Supporting Tools</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>View duty roles inherited by a job role</td>
<td>Manage Duties</td>
<td>• Authorization Policy Manager (APM)</td>
<td>Each logical partition or pillar contains a collection of application roles, and function and data security policies.</td>
</tr>
<tr>
<td>View entitlement or policies carried by a job role</td>
<td>Manage Duties</td>
<td>• APM</td>
<td>In LDAP, the policy store stores application roles and the identity store stores enterprise roles.</td>
</tr>
<tr>
<td>Add a job role to a role hierarchy</td>
<td>Manage Job Roles</td>
<td>• Oracle Identity Management (OIM)</td>
<td>The identity store in LDAP stores enterprise roles.</td>
</tr>
<tr>
<td>Add a duty role to a role hierarchy</td>
<td>Manage Duties</td>
<td>• APM</td>
<td>LDAP stores the role hierarchy and the spanning of roles across multiple pillars or logical partitions.</td>
</tr>
<tr>
<td>Activity</td>
<td>Management Group</td>
<td>System</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>-------------------------------</td>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Create a hierarchy of enterprise (abstract, job, data) roles</td>
<td>Manage Job Roles</td>
<td>OIM</td>
<td>The identity store in LDAP stores enterprise roles.</td>
</tr>
<tr>
<td>Create a hierarchy of (application) duty roles</td>
<td>Manage Duties</td>
<td>APM</td>
<td>The policy store stores duty roles. The identity store stores enterprise roles. Some duty roles may enable actions and their associated users interface features that your enterprise does not want users to perform in Oracle Fusion applications.</td>
</tr>
<tr>
<td>Create a new job role</td>
<td>Manage Job Roles</td>
<td>OIM</td>
<td>The identity store in LDAP stores enterprise roles.</td>
</tr>
<tr>
<td>Change duty roles inherited by a job or abstract role</td>
<td>Manage Duties</td>
<td>APM</td>
<td>The policy store stores duty roles. The identity store stores enterprise roles. Some duty roles may enable actions and their associated users interface features that your enterprise does not want users to perform in Oracle Fusion applications.</td>
</tr>
<tr>
<td>Create a new duty role</td>
<td>Manage Duties</td>
<td>APM</td>
<td>All functions and actions in Oracle Fusion Applications that need to be secured are covered by the reference implementation. In some cases, especially with function customizations, a new duty role may be needed.</td>
</tr>
<tr>
<td>View Segregation of Duties (SOD) policies respected by a duty role</td>
<td>Manage Application Access Controls</td>
<td>Application Access Controls Governor (AACG) in Oracle Enterprise Governance, Risk and Compliance (GRC)</td>
<td>The Security Reference Manuals (SRM) document the segregation of duties (SOD) policies respected within each job role</td>
</tr>
<tr>
<td>View SOD policy violations carried by the duty roles inherited by a job role</td>
<td>Manage Application Access Controls</td>
<td>AACG in GRC</td>
<td>The Security Reference Manuals (SRM) document the SOD policies respected within each job role</td>
</tr>
<tr>
<td>View SOD policy violations</td>
<td>Manage Segregation of Duties Policies</td>
<td>AACG in GRC</td>
<td>The SRM documents the SOD conflicts for each job role</td>
</tr>
<tr>
<td>View the data security policies carried by a job, abstract, and data roles</td>
<td>Manage Data Security Policies</td>
<td>APM</td>
<td>Oracle Fusion Data Security stores data security policies in the policy store. Data security can also be defined in application pages provided by Oracle Middleware Extensions for Applications (FND)</td>
</tr>
<tr>
<td>Create and update HCM security profiles</td>
<td>Manage Data Role and Security Profiles</td>
<td>• Oracle Fusion HCM</td>
<td>This task does not include assigning data roles to the users, which is supported by user provisioning tasks.</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>---------------------------------------</td>
<td>-------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Create (generate) a data role</td>
<td>1. Manage Role Templates</td>
<td>• APM</td>
<td>Data roles are generated automatically based on data role templates and enterprise setup. Changes to data role templates generate new or changed data roles. Create data roles in HCM using the Manage Data Roles and Security Profiles task.</td>
</tr>
<tr>
<td></td>
<td>2. Manage Data Roles and Security Profiles</td>
<td>• Oracle Fusion HCM</td>
<td></td>
</tr>
<tr>
<td>Create a new data security policy (not through generated data roles based on data role templates or HCM security profiles)</td>
<td>Manage Data Security Policies</td>
<td>• APM</td>
<td>Data security can also be defined in application pages provided by Oracle Middleware Extensions for Applications (FND)</td>
</tr>
<tr>
<td>View data role templates defined by a product</td>
<td>Manage Role Templates</td>
<td>• APM</td>
<td></td>
</tr>
<tr>
<td>Create or edit an existing data role template</td>
<td>Manage Role Templates</td>
<td>• APM</td>
<td>Data security can also be defined in application pages provided by Oracle Middleware Extensions for Applications (FND)</td>
</tr>
<tr>
<td>Secure common objects such as attachment categories or profile options</td>
<td>Manage Data Security Policies</td>
<td>• APM</td>
<td>Data security can also be defined in application pages provided by Oracle Middleware Extensions for Applications (FND)</td>
</tr>
<tr>
<td>View, create, update Data Access Sets used to secure Ledgers and Ledger Sets</td>
<td>Manage Data Access Sets</td>
<td>• Oracle Fusion General Ledger</td>
<td></td>
</tr>
<tr>
<td>View, create, update accounting flexfield segment security rules</td>
<td>Manage Security Segments</td>
<td>• Oracle Fusion General Ledger</td>
<td></td>
</tr>
<tr>
<td>View or update the set of job roles that can be provisioned to supplier users</td>
<td>Manage Supplier User Role</td>
<td>• Supplier Portal • Sourcing</td>
<td>These tools are in the Oracle Fusion Procurement product family</td>
</tr>
<tr>
<td>Determine the supplier job roles that the supplier self service administrator can provision to supplier users</td>
<td>Manage Supplier User Role Usages</td>
<td>• Supplier Portal • Sourcing</td>
<td>These tools are in the Oracle Fusion Procurement product family</td>
</tr>
<tr>
<td>Set default supplier job roles based on the set of supplier roles that are defined by performing the Manage Supplier User Roles task</td>
<td>Manage Supplier User Role Usages</td>
<td>• Supplier Portal • Sourcing</td>
<td>These tools are in the Oracle Fusion Procurement product family</td>
</tr>
<tr>
<td>Create a new implementation user</td>
<td>Create Implementation Users</td>
<td>OIM</td>
<td></td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------------------------</td>
<td>-----</td>
<td></td>
</tr>
<tr>
<td>Import legacy users</td>
<td>Import Worker Users</td>
<td>OIM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Import Partner Users</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create a new user</td>
<td>Manage Users</td>
<td>HCM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provision roles to a user</td>
<td>1. Provision Roles to</td>
<td>OIM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Implementation Users</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Manage Users</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

HCM creates a new user and identity when a new worker is created. The Hire Employee and Add Contingent Worker tasks also result in new user creation requests. Creating a new user automatically triggers role provisioning requests based on role provisioning rules.

Implementation users are provisioned through OIM since HCM is not setup at the start of the implementation. The Provision Roles to Implementation Users is not needed once implementation is complete.

Once HCM is setup, HCM is used to provision roles to non-implementation users by performing the Manage Users task. Human Resources (HR) transaction flows such as Hire and Promote also provision roles.

Once supplier users are setup, Supplier Model can be used by internal users to maintain supplier user accounts or supplier users can maintain their accounts in Supplier Portal.
View the job, abstract, and data roles provisioned to a user

1. Manage Users
2. Manage User Principal
3. Provision Roles to Implementation Users

- Human Capital Management (HCM)
- OIM

LDAP stores users, roles and provisioning information.
The Manage User Principal and Provision Roles to Implementation Users tasks are not needed once implementation is complete.

Revoke role from user.

Manage Users

- HCM

You can revoke roles from various Human Resources task flows, the HCM Manage Users task and OIM. User termination includes role revocation.

Approve role provisioning or user account request.

Approve User and Role Provisioning Requests

- OIM

View audit logs

Not applicable

- Oracle Enterprise Manager

Viewing audit logs is a Oracle Fusion Middleware function and not represented by an Oracle Fusion Applications BPM task.

For more information about provisioning identities and configuring audit policies, see the Oracle Fusion Applications Administrator’s Guide.

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

**Define Data Security**

**Data Security: Explained**

By default, users are denied access to all data.

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

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

You secure data by provisioning roles that provide the necessary access. Enterprise roles provide access to data through data security policies defined for the inherited application roles.
When setting up the enterprise with structures such as business units, data roles are automatically generated that inherit job roles based on data role templates. Data roles also can be generated based on HCM security profiles. Data role templates and HCM security profiles enable defining the instance sets specified in data security policies.

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

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

Data is secured by the following means.

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

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

**Data Security Policies**

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

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

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

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

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

---

**Note**

Some data security policies are not defined as grants but directly in applications code. The security reference manuals for Oracle Fusion Applications offerings differentiate between data security policies that define a grant and data security policies defined in Oracle Fusion applications code.

---

A business object participating in a data security policy is the database resource of the policy.

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

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

---

**Important**

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

---

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

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

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

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

**Data Roles**

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

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

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

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

A data role may be granted entitlement over a set people.

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

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

**HCM Security Profiles**

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

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

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.

**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.

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.

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 Oracle Sales Cloud 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 Initiate HCM Spreadsheet Load task. If user and role provisioning rules have been defined, the Initiate HCM Spreadsheet Load process automatically creates user and role provisioning requests as the workers are created.

Once the enterprise is set up, performing the Initiate HCM Spreadsheet Load 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 Initiate HCM Spreadsheet Load task results in new user accounts being created. Worker email addresses as an alternate input for the Initiate HCM Spreadsheet Load 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 Initiate HCM Spreadsheet Load 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 Oracle Sales Cloud.

• Internal employees
• Contingent workers
• External partner contacts
• Partner companies
• Legal entities
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 Initiate HCM Spreadsheet Load 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, Oracle Sales Cloud, or Oracle Fusion Supplier Portal.

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

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

**Manage Duties**

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

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

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

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

**Tip**

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

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

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

**Manage Application Access Controls**

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

**Note**

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

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

**Manage Users**

**Creating Users: Worked Example**

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

**Note**

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

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

<table>
<thead>
<tr>
<th>Prerequisites</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Create a role mapping called All Employees and enter the following conditions.</td>
</tr>
</tbody>
</table>

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

In the role mapping you include the:
- Employee role, and select the **Autoprovision** option
- Expense Claims Administrator role, and select the **Self-requestable** option

**Creating a User**

1. On the Search Person page, click the **Create** icon to open the Create User page.
2. Complete the fields, as shown in this table:

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

3. In the User Details region, leave the User Name field blank. The user name defaults to the user's e-mail ID.
4. In the Employment Information region, select the person type **Employee** and complete the fields as shown in the table:

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

**Assigning Roles to the User**

1. Click **Autoprovision Roles** to provision the employee role to the user.
2. Click **Add Role**.
3. Search for and select the Expense Claims Administrator role.

4. Click Save and Close. The user account is created and the roles are assigned to the user immediately.

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.

What happens when I autopropose roles?

When you autopropose roles to a user, the user’s assignments are reviewed automatically against all current role mappings.

Roles for which the Autopropose option is selected are:

- Provisioned to the user immediately, if the user is eligible for the role and does not currently have it
- Deprovisioned from the user immediately, if the user is no longer eligible for the role but currently has it

You are recommended to autopropose roles to individual users if you know that new or changed role mappings exist for which those users are eligible. Otherwise, roles are not provisioned or deprovisioned automatically until the user’s assignments are next updated.

Where do default user names come from?

By default, user names are defined by Oracle Identity Management (OIM). The user name is typically the user’s first and last names, but this format can be changed in OIM.

The default format of user names can be specified for the enterprise in Oracle Fusion HCM. This setting overrides the OIM default format. Your enterprise may have selected one of the following values:
• Person number
• Party number
• Primary work e-mail

If the default user-name format for the enterprise is the person number, then primary e-mail is used instead for party users who have no person number.

Default user names may not appear for new users until the relevant value (for example, the person number) is available. The user-account request is not submitted to OIM until the value exists.

What happens when I link a user account?

The current person or party record is linked to the selected Oracle Identity Management (OIM) user account. When you click Save, the request is submitted to OIM and the account status is Requested. Once the account status is Active, the user can sign in using the account.

Any roles currently provisioned to the linked account do not appear in the Roles section of the page until the account status is Active. However, you can add roles before clicking Save.

The Link User Account action appears only for persons or party users whose records are not already linked to a user account. When you link a user account, the person or party is not notified automatically. You are recommended to notify the user when the account is linked.
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 Enterprise Governance, Risk and Compliance (GRC) 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 GRC, an access point is the lowest level access for a particular application. In GRC, 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 Oracle Enterprise Governance, Risk and Compliance (GRC) 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 predefine 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.

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

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 Enterprise Governance, Risk and Compliance (GRC) 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 in Oracle Enterprise Governance, Risk and Compliance (GRC). 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 Enterprise Governance, Risk and Compliance (GRC) suite is certified to synchronize with the policy and identity stores for all pillars or partitions of Oracle Fusion Applications and integrated with the Oracle Fusion Applications security approach to roll up entitlements (by means of duty roles) to the roles that are provisioned to internal users. SOD policies can be defined and enforced at any level of authorization. For external users, SOD policies use attribute information stored in the Trading Community Model tables.

**OIM and the SPML Client**

Enterprise business logic may qualify the requester and initiate a role provisioning request by invoking the Services Provisioning Markup Language.
(SPML) client module, as may occur during onboarding of internal users with Human Capital Management (HCM), in which case the SPML client submits an asynchronous SPML call to OIM. Or OIM handles the role request by presenting roles for selection based on associated policies.

OIM recognizes the role provisioning request and initiates a call to AACG.

OIM apprises the SPML client of the current state of the role provisioning request as SOD_CHECK_IN_PROGRESS.

OIM stores the SOD check result as part of OIM audit data.

OIM apprises SPML client of the current state of the SPML request. The provisioning is either still in progress with segregation of duties being checked, or conflicts were found. If conflicts exist, AACG rejects the request and notifies the application.

<table>
<thead>
<tr>
<th>Status</th>
<th>Conflicts</th>
<th>Current State</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOD_CHECK_IN_PROGRESS</td>
<td>Unknown</td>
<td>Request sent to AACG and waiting for response</td>
</tr>
<tr>
<td>SOD_REMEDIATION_IN_PROGRESS</td>
<td>Conflict found</td>
<td>AACG detected violations and remediation is in progress</td>
</tr>
<tr>
<td>SOD_CHECK_APPROVED</td>
<td>No conflict found</td>
<td>No SOD violations found</td>
</tr>
<tr>
<td>SOD_CHECK_REJECTED</td>
<td>Conflict found</td>
<td>AACG detected violations that cannot be remediated</td>
</tr>
<tr>
<td>SOD_REMEDIATION_APPROVED</td>
<td>Conflict found</td>
<td>AACG detected violations that are approved</td>
</tr>
<tr>
<td>SOD_REMEDIATION_REJECTED</td>
<td>Conflict found</td>
<td>AACG detected violations that are rejected by approver</td>
</tr>
</tbody>
</table>

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.
Define Approval Management for Supply Chain Management

Approval Management: Highlights

Use approval management to determine the policies that apply to approval workflows for particular business objects such as expense reports. For example, you can specify levels of approval for expense reports over a particular amount, to reflect your own corporate policies. You also determine the groups of users who act on these workflow tasks, for example, the chain of approvers for expense reports.

Approval management is fully described in the Oracle Fusion Middleware Modeling and Implementation Guide for Oracle Business Process Management. Though the concepts described there apply also to Oracle Fusion Applications, the only setup relevant to Oracle Fusion Applications involves approval groups and task configuration. Customization of approval workflows themselves is described in the Oracle Fusion Applications Extensibility Guide for Developers.

Overview


  See: Introduction to Approval Management

  See: Understanding Approval Management Concepts

Approval Groups and Task Configuration

- An approval group consists of a name and a predefined set of users configured to act on a task in a certain pattern. Refer to the Oracle Fusion Middleware Modeling and Implementation Guide for Oracle Business Process Management.

  See: Administering Approval Groups

See: Using Task Configuration

- To configure a predefined approval policy, select the predefined rule set and click the Edit task icon button.

- To disable a predefined rule set, select the Ignore this participant check box for that rule set.

- To edit the rules within a predefined rule set, you can insert, update, or delete from the seeded rules as needed while in edit mode.

- You can configure a specific rule to automatically approve without being sent to any approver. Modify the routing for that rule so that it is sent to the initiator (which means the requestor is the approver), set the Auto Action Enabled option to True, and enter APPROVE in the Auto Action field.

Customization

- You can optionally customize predefined approval workflows, for example add post-approval activities or additional stages. Refer to the Oracle Fusion Applications Extensibility Guide for Developers.

See: Customizing and Extending SOA Components
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. Help that is embedded in the application, for example hints, can also be customized.

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. For Oracle Cloud, always leave this feature choice selected so that your users can access the Cloud Learning Center.

**Important**

For non-Cloud implementations only: Some help includes links to guides outside the help system. If you select this feature, then these links open guides on the Oracle Technology Network Web site. If you do not select this feature, then your system administrator must download the guides (http://download.oracle.com/docs/cds/E39540_01.zip) and put all the content from within the extracted E39540_01 folder 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. Only users with job roles containing the Application Help Text Administration duty role have access to customize help.

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.

**FAQs for Assign Help Text Administration Duty**

**Who can add and manage custom help?**

Users with the Application 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 Application Help Text Administration duty role on the Role Catalog page, and map additional job roles to this duty role.

Manage Help Security Groups

Creating Help Security Groups: Worked Example

This example demonstrates how to create a help security group to define a set of job roles that have access to help. The help security group can then be assigned to particular help files so that only users with any of the defined roles have access to the help.

The following table summarizes key decisions for this scenario.

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

Define a help security group and assign a duty role to the group.

1. From the Setup and Maintenance work area, find the Manage Help Security Groups task and click Go to Task.
3. Complete the fields, as shown in this table. Leave the start and end dates blank.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help Security Group</td>
<td>HR</td>
</tr>
<tr>
<td>Meaning</td>
<td>HR Only</td>
</tr>
<tr>
<td>Description</td>
<td>Viewing by HR specialists only</td>
</tr>
<tr>
<td>Display Sequence</td>
<td>1</td>
</tr>
</tbody>
</table>

4. Click Save.
5. With your new help security group selected, go to the Associated Roles section and add a new row.
6. Select PER_HUMANRESOURCE_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.
Source Systems: Explained

You can set up source systems to enable users to identify the source of the data they are importing. 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 types of entities the source system contains. For example, you can specify that a source system will contain customer data.

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 to create references between source IDs and the Oracle Sales Cloud database IDs.

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 Sales Cloud 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; however, the source system won’t be able to 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 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 Sales Cloud.

If you do not allow multiple source system references then a record in Oracle Sales Cloud will be created for every source system record. This means that you could potentially create duplicate records in Oracle Sales Cloud.
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.

Map Reports to Work Areas

Additional Report Setup in the Context of the Reports and Analytics Pane: Highlights

Aside from determining which work areas a specific report is mapped to, you can perform additional setup for reports in the context of the Reports and Analytics pane. You can set up report permissions, and enable Oracle Business Intelligence (BI) Publisher reports for scheduled submission.

This additional setup is described in the Oracle Fusion Middleware User’s Guide for Oracle Business Intelligence Enterprise Edition and the Oracle Fusion Applications Extensibility Guide for Business Analysts.
Report Permissions

- You can restrict access to specific reports for specific users, and this security is not limited to the Reports and Analytics pane. Refer to the Oracle Fusion Middleware User’s Guide for Oracle Business Intelligence Enterprise Edition.

See: Assigning Permissions

Oracle Business Intelligence Publisher Reports Submission

- Oracle BI Publisher reports must be registered as processes with Oracle Enterprise Scheduler to be enabled for scheduling. This registration also enables a Schedule link for the report in the Reports and Analytics Pane. Refer to the Oracle Fusion Applications Extensibility Guide for Business Analysts, and perform the following steps in the specified order.
  - Create an Oracle Enterprise Scheduler job definition for the report.
  - Specify the job definition details in the report’s properties.

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 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.

For more information on configuring custom objects, see the Oracle Sales 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

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 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.

**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

### Define Reference Data Sharing

#### Reference Data Sets: Explained

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

Consider the following scenario.

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

#### Partitioning

The partitioning of reference data and creation of data sets enable you to create reference entities across tables or lookup types, and share modular information and data processing options among business units. With the help of partitioning, you can choose to create separate sets and subsets for each business unit depending upon its business requirement, or create common sets or subsets to enable sharing reference data between several business units, without the need for duplicating the reference data. Partitioning provides you the flexibility to handle the reference data in a way appropriate to your business needs.
The following figure illustrates the reference data sharing method (assignment to one set only, with common values) where the user can access the data assigned to a specific set in a particular business unit, as well as access the data assigned to the common set.

Reference Data Sets and Sharing Methods: Explained

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

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

Reference Data Sets

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

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

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

**Reference Data Sharing Methods**

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

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

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

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

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

**Assigning Reference Data Sets to Reference Objects: Points to Consider**

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

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

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

**Determinant**

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

**Reference Groups**

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

**Note**

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

**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.

### Define ISO Reference Data

#### Euro Currency Derivation: Explained

Use the Derivation Type, Derivation Factor, and Derivation Effective Date fields to define the relationship between the official currency (Euro) of the European Monetary Union (EMU) and the national currencies of EMU member states. For each EMU currency, you define its Euro-to-EMU fixed conversion rate and the effective starting date.

**Note**
If you need to use a different currency code for Euro, you can disable the predefined Euro currency and create a new one.

#### Derivation Type

The **Euro currency** derivation type is used only for the Euro, and the **Euro derived** derivation type identifies national currencies of EMU member states. All other currencies do not have derivation types.

#### Derivation Factor

The derivation factor is the fixed conversion rate by which you multiply one Euro to derive the equivalent EMU currency amount. The Euro currency itself should not have a derivation factor.
Derivation Effective Date

The derivation effective date is the date on which the relationship between the EMU currency and the Euro begins.

Natural Languages: Points to Consider

Natural languages are all the languages that humans use, written and spoken. If a language is enabled, then users can associate it with entities, for example as languages spoken by sales representatives. When managing natural languages, consider tasks to perform and best practices for entering particular values.

Tasks

Once you add a language, it cannot be deleted, just disabled. You can optionally associate natural languages with International Organization for Standardization (ISO) languages and territories, just for reference.

Values

When you create a natural language, use the alpha-2 ISO code as the language code, or, if not available, then alpha-3. If the language is not an ISO language, then use x- as a prefix for the code, for example x-ja for a Japanese dialect. Use the sgn code of ISO-639-2 for sign languages, followed by territory code, for example sgn-US for American Sign Language. You can also use Internet Assigned Numbers Authority (IANA) language tags.

The natural language description should be the language name with territory name in parenthesis where needed, for example English (Australia) and English (Canada).

Defining Currencies: Points to Consider

When creating or editing currencies, consider these points relevant to entering the currency code, date range, or symbol for the currency.

Currency Codes

You cannot change a currency code after you enable the currency, even if you later disable that currency.

Date Ranges

Users can enter transactions denominated in the currency only for the dates within the specified range. If you do not enter a start date, then the currency is
valid immediately. If you do not enter an end date, then the currency is valid indefinitely.

Symbols

Even if you enter a symbol for a currency, the symbol is not always displayed when an amount is displayed in this currency. Some applications use currency symbols when displaying amounts. Others, like Oracle Fusion General Ledger, do not.

Manage Data Security Policies

Data Security in the Security Reference Implementation: Explained

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

The security implementation of an enterprise is likely a subset of the reference implementation, with the enterprise specifics of duty roles, data security policies, and HCM security profiles provided by the enterprise.

The business objects registered as secure in the reference implementation are database tables and views.

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

Data Security Policies in the Security Reference Implementation

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

Tip

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

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

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

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

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

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

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

**Data Roles**

The security reference implementation includes no predefined data roles to ensure a fully secured initial Oracle Fusion Applications environment.

The security reference implementation includes data role templates that you can use to generate a set of data roles with entitlement to perform predefined business functions within data dimensions such as business unit. Oracle Fusion Payables invoicing and expense management are examples of predefined business functions. Accounts Payable Manager - US is a data role you might generate from a predefined data role template for payables invoicing if you set up a business unit called US.

HCM provides a mechanism for generating HCM related data roles.

**Data Security: Explained**

By default, users are denied access to all data.

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

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

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

When setting up the enterprise with structures such as business units, data roles are automatically generated that inherit job roles based on data role templates. Data roles also can be generated based on HCM security profiles. Data role templates and HCM security profiles enable defining the instance sets specified in data security policies.
When you provision a job role to a user, the job role implicitly limits data access based on the data security policies of the inherited duty roles. When you provision a data role to a user, the data role explicitly limits the data access of the inherited job role to a dimension of data.

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

Data is secured by the following means.

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

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

**Data Security Policies**

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

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

<table>
<thead>
<tr>
<th>Who</th>
<th>can do</th>
<th>what</th>
<th>on which set of data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounts payable managers</td>
<td>view</td>
<td>AP disbursements</td>
<td>for their business unit</td>
</tr>
</tbody>
</table>
A data security policy is a statement in a natural language, such as English, that typically defines the grant by which a role secures business objects. The grant records the following.

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

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

**Note**

Some data security policies are not defined as grants but directly in applications code. The security reference manuals for Oracle Fusion Applications offerings differentiate between data security policies that define a grant and data security policies defined in Oracle Fusion applications code.

A business object participating in a data security policy is the database resource of the policy.

Data security policies that use job or duty roles refer to data security entitlement. For example, the data security policy for the Accounts Payable Manager job role refers to the view action on AP disbursements as the data security entitlement.

**Important**

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

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

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

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

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

Data roles are implemented as job roles for a defined set of data. A data role defines a dimension of data within which a job is performed. The data role inherits the job role that describes the job. For example, a data role entitles a user to perform a job in a business unit.

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

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

A data role may be granted entitlement over a set people. For example, a Benefits Administrator A-E is allowed to administer benefits for all people that have a surname that begins with A-E.

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

HCM Security Profiles

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

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

Applications outside of HCM can use the HCM Data Roles UI pages to give their roles access to HR people.

Masking and Encryption

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

Database Resources and Data Security Policies: How They Work Together

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

The database resource defines and instance of a data object. The data object is a table, view, or flexfield.

The following figure shows the database resource definition as the means by which a data security policy secures a data object. The database resource names the data object. The data security policy grants to a role access to that database resource based on the policy’s action and condition.

Database Resources

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

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

Data Security Policies

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

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

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

**Actions**

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

**Conditions**

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

**Tip**

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

---

**Securing Data Access: Points to Consider**

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

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

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

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

- Specific rows of data (data object) identified by primary key
- Groups of data (instance set) based on a predicate that names a particular parameter
- Data objects or instance sets based on runtime user session variables
Data is either identified by the primary key value of the row in the table where the data is stored. Or data is identified by a rule (SQL predicate) applied to the WHERE clause of a query against the table where the data is stored.

Grants

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

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

Grants control which data a user can access.

Note

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

A grant logically joins a user or role and an entitlement with a static or parameterized object instance set. For example, `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.

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

#### Menu Customization: Explained

You use the Manage Menu Customizations task to customize the navigator and home page menus. This task is available from the Setup and Maintenance work area, which is accessible from the Administration menu in the Oracle Fusion Applications global area. Select either **Customize - Navigator** or **Customize - Homepage** to proceed with the customization activity.

---

**Note**

To perform menu customization at run time, it is important that you have the required privileges.

---

You customize the menus at the site level and your changes affect all users (or all users of a tenant if in a multi-tenant environment).

---

**Tip**

If you are making minor changes, such as adding or editing one or two nodes, then you can hide the changes until you have completed your customizations. However, if you are making more than minor changes, such as rearranging several nodes, you might want to instead create a sandbox before customizing menus.

---

### Navigator Menu Configuration

The navigator menu is the global menu that is accessible from the Oracle Fusion Applications global area. It allows users to navigate directly to the pages inside Oracle Fusion Applications as well as to outside web pages. The menu is composed of links (items) that are organized in a hierarchy of groups.
You can customize the navigator menu to address needs that are specific to your organization. For example, you might want to add specialized groupings for cross-functional teams or add links to web pages or external applications. You can add groups and links to the navigator menu, as well as hide and show them.

The Manage Menu Customizations task displays the menu groups as expandable nodes, with which you can traverse the menu hierarchy.

**Note**

Not all Oracle Fusion Applications pages appear in the navigator menu, because some pages are accessible from a work area or from other links in the global area such as the **Home** link.

The following table lists the Navigator menu customization tasks that you can perform at run time as well as the tasks that you cannot perform.

<table>
<thead>
<tr>
<th>Permitted Tasks</th>
<th>Restricted Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Add and delete custom groups.</td>
<td>• You cannot add menu items (links) as top-level nodes. You can add nodes to only the groups in the top level and subgroups.</td>
</tr>
<tr>
<td>• Edit any group.</td>
<td>• You cannot delete nodes that are delivered with the product. Instead, you can hide them.</td>
</tr>
<tr>
<td>• Add and delete custom items.</td>
<td>• You cannot move nodes. Instead, you must duplicate the node and hide the original node.</td>
</tr>
<tr>
<td>• Edit any item.</td>
<td></td>
</tr>
<tr>
<td>• Specify navigation for an item:</td>
<td></td>
</tr>
<tr>
<td>• Specify navigation to a UI Shell page in an Oracle Fusion application.</td>
<td></td>
</tr>
<tr>
<td>• Specify navigation to an external web page.</td>
<td></td>
</tr>
<tr>
<td>• Hide or show groups and items.</td>
<td></td>
</tr>
</tbody>
</table>

**Home Page Menu Configuration**

The home page menu is the set of tabs that are displayed in the Oracle Fusion Applications global area. The home page menu displays tabs for all the items in the menu for which the end user has access privileges. You can add tabs to the home page menu, as well as hide and show them.

The following table lists the Home page menu customization tasks that you can perform at run time as well as the tasks that you cannot perform.

<table>
<thead>
<tr>
<th>Permitted Tasks</th>
<th>Restricted Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Add and delete custom items.</td>
<td>• You cannot add menu items (links) as sub-nodes. All nodes are top-level nodes.</td>
</tr>
<tr>
<td>• Edit any item.</td>
<td>• You cannot delete nodes that are delivered with the product. Instead, you can hide them.</td>
</tr>
<tr>
<td>• Specify navigation to a UI Shell page in an Oracle Fusion application.</td>
<td>• You cannot move nodes. Instead, you must duplicate the node and hide the original node.</td>
</tr>
<tr>
<td>• Hide or show items.</td>
<td></td>
</tr>
</tbody>
</table>
Adding Navigator Menu Group: Points to Consider

You arrange the navigator menu by building a hierarchy of nested groups.

Use the View menu to expand or collapse a group of nodes. You can also right-click a node and access similar actions to facilitate tree navigation.

Adding Groups

To add a group, you can insert a group above or below a peer group or insert a child group. You edit a group by defining a label and specifying whether the group should be rendered. You typically hide the group until all changes have been completed.

Adding Menu Items: Points to Consider

The home page menu items are URL links to home pages in Oracle Fusion applications. The Navigator menu items can either be links to UI Shell pages or links to external applications and web sites.

In the menu hierarchy, the home page menu items are always top-level items. Whereas, you can add Navigator menu items to top-level groups and to their subgroups but you cannot add navigator menu items as top-level nodes.

Adding a Home Page Menu Item

To add a home page menu item, navigate to the place where you want the item to appear and insert it above or below the existing item. You can also duplicate an existing menu item and position it at the required location. You must provide a label for the menu and link the menu item to a UI Shell page.

Adding a Navigator Menu Item

To add a Navigator menu item, you navigate to the item’s group and insert the item above or below another item.

You can also duplicate an existing item. You must provide a label for the menu and either link the menu item to a UI Shell page or link it to an external web site or application.

You can link a Navigator menu item to the following:

- A UI Shell page in an Oracle Fusion application.
- A dynamic URL of a page outside of Oracle Fusion Applications where the host, port, or context root might change.
- A Static URL of a page outside of Oracle Fusion Applications where the host, port, or context root does not change.
Linking to a UI Shell Page

If the new item points to a UI Shell page in an application, then you must provide the name of the web application and the view ID of the target page. The web application name and view ID can be obtained from an existing menu item that links to the same UI Shell page.

In a non-Cloud implementation, you also can obtain the web application name from the context root for the application, and you can obtain the view ID from the id attribute for the page's <view> tag in the product's public_html/WEB-INF/adfc-config.xml file.

If you want secure access to the target UI Shell page from the menu item, then you must provide the name of the secured resource and the name of the policy store's application stripe. When an end user clicks the link, the Oracle Fusion Applications checks the secured resource and the Lightweight Directory Access Protocol (LDAP) policy store to determine whether the user has the privilege to view the page.

If there is another menu item that points to the same page, then you can get the secured resource name and application stripe from that item. In a non-Cloud implementation, you also can obtain the application stripe from the jps.policystore.applicationid parameter in the application's weblogic-application.xml file. Examples of application stripes are crm, fscm, and hcm.

For non-Cloud applications, you can determine the secured resource name by obtaining the name of the web page's page definition file. By default, the page definition files are located in the view.PageDefs package in the Application Sources directory of the view project. If the corresponding JavaServer Faces (JSF) page is saved to a directory other than the default (public_html), or to a subdirectory of the default, then the page definition will also be saved to a package of the same name. An example of a secured resource name is oracle.apps.view.pageDefs.CaseList_Form_Attach_UShellPagePageDef.

A UI Shell page might take parameters and display or act differently based on the parameters that are passed in. For example, if accessing a page from one group in the menu hierarchy, the parameter might be set to status=Open and if accessing the page from a different group, the parameter might be set to status=Closed. If the page takes parameters, you can use the Page Parameters List text box to provide a semicolon-delimited string of name-value pairs, such as org=m1;context=s1. You can use expression language (EL) to specify the parameters. If the EL evaluates to an Object, the toString value of that Object is passed as the value of the parameter.

Linking to the Dynamic URL of an External Web Site or Application

Linking a menu item to a dynamic URL is beneficial in cases where the host, port, or context root to which you point frequently changes. Instead of updating the link to each application, you can update the details of the web application in the topology registration, and that change affects all menu items that contain dynamic links pointing to that web application. For example, you would need a dynamic URL to link to a test version of an application and you will need to change the host and port when you move the application from a test environment to a production environment.
To link to a page outside of Oracle Fusion Applications where the host, port, or context root might change, you must first register the web application in the topology using the Register Enterprise Applications task.

While creating a new menu item on the Create Item Node dialog box, select the Dynamic URL option and provide the details of the web application as per the following example.

When the complete URL to be linked is: http://example:9011/myApp/faces/Page1,

- The name of the web application added to topology would be: myApp (the value that would eventually appear in the Web Application list) and the protocol host, port, and context root values of the URL would be: http://example:9011/myApp
- The value to be provided in the Destination for Web Application field would be: /faces/Page1

Once the menu item is linked to the dynamic URL, the target page appears in a new browser window or tab when you click the menu item.

**Linking to a Static URL of an External Web Site or Application**

This option is used when you link a menu item to a page outside Oracle Fusion Applications where the host, port, or context root remains constant. For example, you can use a static URL to link to http://www.oracle.com.

**Hiding or Displaying Menu Nodes: Points to Consider**

While you are creating or working with a menu group or a menu item, you might want to prevent end users from accessing the node. You can hide the menu group or menu item while you are working with it, and then show the node when you have completed the task.

**Working with Nodes**

The Manage Menu Customizations page shows all nodes. The Rendered check box is selected by default for all nodes that are added and are visible.

To hide a node, clear the Rendered check box. You can edit the node anytime to either display or hide it.

If you want a menu group or a menu item to appear only if certain conditions are met, you can use an expression language (EL) command to make the node to appear. For example, #{securityContext.userInRole('ADMIN')}.

A node that appears in italics either contains an EL command or the Rendered check box beside it was cleared, and therefore is hidden from end users.

**Tip**

For major changes that need to be tested and approved, you might want to use the sandbox manager instead of hiding and showing nodes.
Design Time Menu Customizations: Highlights

The menu customization feature provides several options to add, modify, and organize the Navigator and home page menus during design time. You must have developer rights to perform these customizations.

Note

Design time menu customizations are not applicable to Oracle Cloud implementations.

An overview of customizing the Navigator menu and home page is provided in the Oracle Fusion Applications Extensibility Guide.

Customizations

- Use Oracle JDeveloper to customize the Navigator and home page menus at design time.
  See: Customizing Menus

- Define translations for your customizations in the locales you support.
  See: Translating Menu Customizations

- Customize the page template to display the Navigator menu groups as separate menus, each of them displaying their list of menu items. Refer to the Oracle Fusion Applications Developer's Guide.
  See: Rendering the Navigator Menu as Dropdown Buttons

Troubleshooting Navigator Menu: Highlights

If the Navigator menu does not display customizations as expected, use the following troubleshooting tips to verify the changes.

Issues and Resolutions

- If an expected menu item does not appear in the Navigator menu, verify whether the menu item has been hidden from view.

- If a custom menu item was added and the browser does not display the page indicated by the URL, open the Manage Menu Customizations task and verify whether the web application name is the same as the context root for the application, and that the view ID is the id attribute for the page's <view> tag in the product's public_html/WEB-INF/adfc-config.xml file. The URL should not contain the .JSPX suffix.

- If you see a "webApp value not define" error message when you choose an item in the Navigator menu, verify whether the application is in the topology tables. Refer to the Oracle Fusion Applications Administrator's Guide.
Managing Audit Policies: Explained

Auditing is used to monitor user activity and all configuration, security, and data changes that have been made to an application. Auditing involves recording and retrieving information pertaining to the creation, modification, and removal of business objects. All actions performed on the business objects and the modified values are also recorded. The audit information is stored without any intervention of the user or any explicit user action.

Use audit policies to select specific business objects and attributes to be audited. The decision to create policies usually depends on the type of information to be audited and to the level of detail that is required to be reported.

Enabling Audit Functionality

To enable audit, ensure that you have administrative privileges. For Oracle Fusion Applications, you must configure the business objects and select the attributes before enabling audit. If you enable audit without configuring the business objects, auditing remains inactive. By default, auditing is disabled for all applications.

To enable auditing for Oracle Fusion Middleware products, select one of the levels at which auditing is required for that product. The audit levels are predefined and contain the metadata and events to be audited. For more information, refer to the Oracle Fusion Middleware documentation and also the Oracle Enterprise Repository for Oracle Fusion Applications at http://fusionappsoer.oracle.com.

If you do not want an application to be audited, you can stop the audit process by setting the Audit Level option to None. While viewing the audit report for that application, you can specify the period during which auditing remained enabled.

Configuring Audit Business Object Attributes: Points to Consider

Audit allows you to track the change history of particular attributes of a business object. However, those objects and their attributes must be selected for audit and auditing must be enabled for that application. Your configuration settings determine which attributes to audit for a given object, and when the audit starts and ends. Auditing takes into account all the create or insert, update, and delete operations performed on an object and its attributes.
To configure audit business object attributes, navigate to the Manage Audit Policies page in the Setup and Maintenance work area.

**Selecting an Application**

To set up auditing, you must select a web application that contains the required business objects that can be audited. From the list of business objects, select those business object that you want to audit. Selecting a business object also displays its attributes that are enabled for auditing.

**Selecting Attributes**

For each selected business object to be audited, select the corresponding attributes to include in the audit. All attributes that belong to that object are by default selected for audit and appear on the user interface. However, you can add or remove attributes from the list. When you remove an attribute from the list, you stop auditing it even when the parent object is selected for audit. So, if you want an attribute to be audited, you must add it to the list.

**Note**

If the object selected in an audit hierarchy is also a part of several other audit hierarchies, the attribute configuration for that object is applicable to all the hierarchies in that application.

**Starting and Stopping Audit**

The business object is ready for audit after you select its attributes and save the configuration changes. However, to start auditing, the audit level for Oracle Fusion Applications must be set to **Auditing** on the Manage Audit Policies page.

To stop auditing an object, you can deselect the entire object and save the configuration. As a result, all its selected attributes are automatically deselected and are not audited. To continue to audit the business object with select attributes, deselect those attributes that are not to be audited.

When end-users view the audit history for an application, they can specify the period for which they want the results. Therefore, it is important to note when you start and stop auditing an application. For example, today if end-users intend to view the audit history of an object for the previous week, but auditing for that object was stopped last month, they would not get any audit results for that week because during the entire month that object was not audited. Even if you enable audit for that object today, end-users cannot get the wanted results because audit data until today is not available.

**Configuring Audit: Highlights**

You can set up auditing for Oracle Fusion Applications using the Manage Audit Policies page in the Setup and Maintenance work area of Oracle Fusion Applications.
To set up auditing for Oracle Fusion Middleware products, you must select the level of auditing that maps to a predefined set of metadata and events that have to be audited. Information on configuring audit for Oracle Fusion Middleware products is provided in Oracle Fusion Middleware guides.

You can also create a configuration file and deploy it to audit a specific Oracle Fusion Middleware product. The configuration details for Oracle Fusion Middleware products are available in the form of audit-specific assets that can be used to create the configuration file (config.xml). For more information, see the Oracle Enterprise Repository for Oracle Fusion Applications at http://fusionappsore.oracle.com, and search with Audit as the Asset Type to get the list of audit-specific assets.

**Oracle Fusion Middleware Products**


  See: Auditing Web Services

**Oracle Fusion Security Products**

- Configure business objects to enable auditing in Oracle Fusion security products. Refer to Oracle Fusion Middleware Application Security Guide.

  See: Oracle Fusion Middleware Audit Framework Reference

---

**Manage Oracle Social Network Objects**

**Managing Oracle Social Network Objects: Explained**

Use the **Manage Oracle Social Network Objects** task for managing the Oracle Social Network Objects. The integration of Oracle Social Network Cloud Service with applications and business processes brings key attributes from the applications to share, socialize, and update information. This helps in making better business decisions based on additional information that you obtain and analyze within your social network environment.

Use the **Manage Oracle Social Network Objects** page to set up and define:

- The business objects and attributes to enable

- The enablement method for social network integration with Oracle Fusion Applications

You can access the **Manage Oracle Social Network Objects** page by starting in the Setup and Maintenance Overview page and searching for the task named **Manage Oracle Social Network Objects**.

Use social network to:

- Discuss projects and plans in public forums
• Maintain:
  • Membership groups
  • Activity feeds of the people you choose
• Facilitate:
  • One-on-one Conversations
  • Reviews
  • Document sharing

Note
Oracle Social Network Cloud Service is currently available in Cloud implementations only.

An important aspect of managing Oracle Social Network objects is enabling business objects for integration.

Enabling Business Objects for Integration
A business object can’t be shared within social network until a functional administrator or implementor:

  • Accesses the Manage Oracle Social Network Objects page in Oracle Fusion Applications
  • Enables the business object for social network integration

Options for Enabling Oracle Social Network Objects: Explained

To enable business objects and apply attributes for Oracle Social Network Cloud Service integration with Oracle Fusion Applications, use the Manage Oracle Social Network Objects task.

In the Manage Oracle Social Network Objects page, you can:

  • Enable an object
  • Disable an object
  • Enable all objects
  • Enable business object attributes

To access the Manage Oracle Social Network Objects page:

1. Search for the Manage Oracle Social Network Objects task in the Setup and Maintenance work area.
2. In the Search Results section, click the Go to Task icon to open the Manage Oracle Social Network Objects page.

Note
Custom objects and attributes created in Application Composer in the mainline are also displayed on the Manage Oracle Social Network Objects page. You can enable these objects and attributes for social network integration.

Enable Object

To enable a business object:

1. Access the Manage Oracle Social Network Objects page.
2. In the Business Objects section, select a business object, click Enable Object, and select one of the enablement options. The business objects are grouped by modules. The available enablement options are:
   - Manual: (Recommended) Empowers the user to decide whether to share each instance of the object with social network.

Note

Once shared, all updates to the enabled attributes of the object instance, and deletes, are sent to social network. Updates to attributes that are not enabled are not sent.

- Automatic: Automatically sends the newly enabled object instances and updates to social network.

Note

All object instances are automatically shared with social network upon creation, and all subsequent updates to the enabled attributes of the object instances, and deletes, are automatically sent to social network.

- No: Does not send any information on object instance to social network. This is the default option.

3. Click OK.

This enables the selected business object, and empowers the user to decide whether to share each instance of the object with social network.

Note

After you enable an object, you must enable one or more attributes in the Attributes section of the Manage Oracle Social Network Objects page. Updates to enabled attributes are sent to social network.

Disable Object

To disable a business object:

1. Access the Manage Oracle Social Network Objects page.
2. In the Business Objects section, select a business object, and click Disable Object.
3. Save your changes.

This disables the selected business object by updating the enablement option as No.

Enable All

To enable all business objects:

1. Access the Manage Oracle Social Network Objects page.
2. In the Business Objects section, click Enable All.
3. Save your changes.

This enables all business objects in bulk, and updates the enablement option of all business objects as Manual.

Note

- After you enable business objects, you must enable one or more attributes in the Attributes section of the Manage Oracle Social Network Objects page. Updates to the enabled attributes are sent to social network.
- If you enable a business object, but don’t configure any attributes for the enabled business object, no attributes are sent to social network during create and update. The only exception is that some internal bookkeeping information are sent. Deletes are sent as usual.

Status Column

The Status column in the Business Objects table visually indicates:

- Whether a business object is enabled
- Which enabled business objects don’t yet have an enabled attribute assigned

The status indicators include:

- A check mark, which indicates that you have configured attributes for an enabled business object
- A warning sign, which indicates that you have not configured any attributes for an enabled business object

Enable Business Object Attributes

To enable business object attributes:

1. In the Attributes section, click Add to display the Select Attributes dialog where you can select attributes to add to the table.
2. Select an attribute name in the table, and select the Enabled check box to enable the attribute.
3. Click OK.
4. Save your changes.

In the Attributes table, you can also:
• Click **View** to view a list of all attributes that are enabled.
• Click **Remove** to remove attributes from the table.
• Hover over the **Attribute Information** icon displayed next to descriptive flexfield attributes to view information about the attributes.

**Update Translations: Explained**

The Update Translations process sends attribute labels and business object names to Oracle Social Network Cloud Service for use in the user interface.

In social network, the attribute or business object labels appear in the language of your locale. If you change the locale in social network, then the attribute or business object labels appear in the updated language. However, the data appears in the language in which it was originally sent to social network. If you have previously sent an instance of the business object to social network, then the instance data isn’t updated. Clicking the **Update Translations** button on the **Manage Oracle Social Network Objects** page sends translations for business objects with the enablement option as **Manual** or **Automatic**.

**Synchronize Business Objects: Explained**

Use the **Synchronize** button on the **Manage Oracle Social Network Objects** page to synchronize business objects. This re-sends the definitions of business objects having the enablement option as **Manual** or **Automatic** to Oracle Social Network Cloud Service.

Use the Synchronize button at the:

• **Business Objects table level**: To re-send the definitions of a selected business object to social network. This button is enabled only when you select a row for a business object with the enablement option as **Manual** or **Automatic**.

• **Manage Oracle Social Network Objects page level**: To re-send the definitions of all business objects with the enablement option as **Manual** or **Automatic** to social network.

**Note**

If you had modified any business object enabled for social network and not saved your changes, then on clicking the **Synchronize** button, a warning message appears. This message informs you that you have not saved your changes, and you can select one of the following options:

• **Save and Synchronize**: To save the modified business objects, and synchronize the unmodified business objects.

• **Synchronize**: To ignore any unsaved business objects, and only synchronize the unmodified business objects.

• **Cancel**: To cancel the synchronization task.
FAQs for Manage Oracle Social Network Objects

What happens if I update translations?

When you update translations, you send translations for business objects with the enablement option as **Manual** or **Automatic** to Oracle Social Network Cloud Service.

On updating translations, you also:

- Synchronize the newly translated text from Oracle Fusion Applications so that it can be used within social network. This means you can:
  - Install and enable a new language.
  - Take a language patch at any time.
  - Send attribute labels and business object names to social network for use in its user interface.

How can I update translations?

Use the **Update Translations** button on the **Manage Oracle Social Network Objects** page for subsequent updates to labels and attributes.

Use the **Update Translations** button at the:

- **Business Objects table level**: To send translations for a selected business object to Oracle Social Network Cloud Service. This button is enabled only when you select a row for a business object with the enablement option as **Manual** or **Automatic**.

- **Manage Oracle Social Network Objects page level**: To send translations for all business objects with the enablement option as **Manual** or **Automatic** to social network.

---

**Note**

When you save the enablement of a business object to social network, it sends the translations as well. Hence, you need not click the **Update Translations** button after saving the enablement.

---

When do I update translations?

Run the **Update Translations** process only after you install a new language pack of Oracle Fusion Applications.

Updating translations synchronizes the newly translated text to Oracle Social Network Cloud Service for integration with Oracle Fusion Applications.

---

**Note**
When you save the enablement of a business object to social network, it sends
the translations as well. Hence, you need not click the Update Translations
button after saving the enablement.

What happens if I synchronize business objects?

When you synchronize business objects, you re-send the definitions of business
objects having the enablement option as Manual or Automatic to Oracle Social
Network Cloud Service.

When do I synchronize business objects?

Run the Synchronize process after you use customization sets to import the setup
from the Manage Oracle Social Network Objects page in another environment.
You can also run the process any time you want to synchronize the settings of
business objects with Oracle Social Network Cloud Service without making any
changes in the Manage Oracle Social Network objects page.

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. To create or edit a message, navigate to the Manage Messages page in the Setup and Maintenance work area.

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 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.

FAQs for Manage Applications Core Administrator Profile Values

How can I enable the privacy statement?

Use the Privacy Statement URL profile option to enable the Privacy Statement menu item in the global area. This menu item in the Settings and Actions menu is disabled by default.

Open the Setup and Maintenance work area, and use the Manage Applications Core Administrator Profile Values task to find the Privacy Statement URL profile option. In the Profile Value column, enter the full URL of the web page that contains the privacy content you want the menu item to link to.
You can use Oracle Sales Cloud' Computer Telephony Integration (CTI) to place a call to a contact from a hyperlink on the phone number or phone icon.

Here are a few topics that are important to know when using CTI:

- Normal call flow
- Interaction Records and Notes
- Operational Notes

**Note**
CTI must be enabled to make calls using the various contact information pages and pop-up UIs. When enabled, phone numbers appear as hyperlinks. Interaction logging is available if that feature is enabled. If interaction logging is available, a note indicating that fact will be displayed.

**Normal Call Flow**
CTI uses a call-the-caller-then-call-the-callee procedure for completing a phone call. That format and the normal flow of this procedure are described below.

- **You initiate a call**
  If you see a small orange square next to a contact or customer name, click the square to display further details, including phone numbers. To place a call, place your mouse over the phone number hyperlink and click.

**Note**
CTI does not work on phone numbers that are marked with a Do Not Call icon.

- **Select a Calling Phone**
  Choose the calling phone number. Usually the calling phone is a number from your profile information. Alternately, if you need to use a phone not in your profile, you can specify a different number to originate your call.
• Call Flow

After you select the calling phone number, the system calls you back on that number, waits for you to answer, and then calls the person for whom the call is intended.

Interaction Records and Notes

CTI automatically creates an interaction record of the call, when that feature is enabled. The details window that provides the phone number may also show an Interaction icon that you can click to display a list of interaction records to edit, for example to provide a description of the call. The window may also provide a notes feature that you can use to record notes during the call.

Interaction Logging

The interaction record is logged as soon as the call is either successfully set up or known to have failed.

The interaction log records the customer, call participants, a timestamp noting the start time of the call, the direction of the communication, in or outbound, and the resolution code. The description is automatically updated with these three items:

• Call ID from OWLCS
• Your chosen phone number

• Contact phone number

The call resolution code is determined from OWLCS and recorded in the interaction:

<table>
<thead>
<tr>
<th>OWLCS Call Status</th>
<th>Resolution Code in Interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>CallConnected</td>
<td>CALL ANSWERED</td>
</tr>
<tr>
<td>CallAborted</td>
<td>FAILED</td>
</tr>
<tr>
<td>CallHangUp</td>
<td>FAILED</td>
</tr>
<tr>
<td>CalledPartyBusy</td>
<td>NOT AVAILABLE</td>
</tr>
<tr>
<td>CalledPartyNoAnswer</td>
<td>NOT AVAILABLE</td>
</tr>
<tr>
<td>CalledPartyNotReachable</td>
<td>NOT AVAILABLE</td>
</tr>
<tr>
<td>CallingPartyBusy</td>
<td>FAILED</td>
</tr>
<tr>
<td>CallingPartyNoAnswer</td>
<td>FAILED</td>
</tr>
<tr>
<td>CallingPartyNotReachable</td>
<td>FAILED</td>
</tr>
</tbody>
</table>

Editing interactions

Once the call is established, if Interactions is available, you can use the Interactions icon on the UI to launch the interaction record list view. Select the current interaction record to edit it.

Operational Notes

Because of the call-the-caller-then-call-the-callee format, there are some conditions that may occur due to several calling situations. Some of these conditions are described below:

• Why don’t I hear a ring-back tone? As soon as you answer the system call-back, the system immediately dials the contact. You won’t hear a ring-back tone as in a normal outbound phone call. However, you can tell that the call attempt is progressing because:

  • The phone indicates that the connection is active. If the call to the contact reaches a busy tone or the call attempt times out, the connection is dropped.

  • The dialing window stays on the screen while the call attempt is progressing. It disappears when the connection is either successfully established or fails.

• What if your phone is busy and the call-back goes directly to voice mail? Normally this would not happen because you would not initiate a new call when you are already busy on another call. However, this situation could occur due to a race condition, that is where another incoming call reaches your phone before the CTI call-back. When this happens, two different scenarios could occur:

  • If your phone is configured for busy-forward-all-to-voice-mail, the CTI call would be forwarded to your voice mail, and the system thinks that
the caller has answered the call and will proceed to call the contact. On answering, the contact hears your voice-mail greeting.

- If your phone is capable of presenting a second call to the user, as is supported by many office phones and mobile phones, then you can still answer the CTI call and there is no issue.

- What if you wait too long to answer the call-back? In other words, you wait longer than the ring-no-answer-forward-to-voice-mail timer on the phone system and the call goes to voice mail. Normally, this would not happen because you are expecting the inbound call after you started the call, and would answer promptly. However, if for some reason you do not answer and allow the call to ring-no-answer-forward to voice mail, then the system would think that you have answered the call and will proceed to call the contact. On answering, the contact hears your voice-mail greeting.

- What if the contact does not answer in 30 seconds and the system abandons the call attempt? If the contact’s voice mail is configured to answer after 30 seconds, you will not be able to leave a message.

**Oracle Sales Cloud CTI: Top Tasks**

Oracle Sales Cloud Computer Telephony Integration (CTI) is a feature of the customer contact process. Phone communication to customers and employees is initiated with a click of the mouse, leveraging your customer contact information and the application context. The CTI feature uses Oracle WebLogic Communication Services, OWLCS, to enable communications. Applications that provide the CTI functionality do so primarily through contextual actions.

Additionally, CTI utilizes Oracle Sales Cloud interactions as an optional transaction logging feature that will track information about the call such as the customer, call participants, a timestamp noting the start time of the call, the direction of the communication, in or outbound, and the resolution code.

CTI integrates with your telephony environment and must be manually enabled in your deployment. This topic highlights what is required to set up the CTI feature and to implement logging of the calls made using the CTI feature.

Terms used in setting up these communications

- PSTN: Public switched telephone network is the network of the world’s public circuit-switched telephone networks.
- SIP: Session initiation protocol, an open signaling protocol standard that is used to set up phone calls
- TPCC: Third Party Call Control enables an application to control the telephony network to set up calls automatically.
- OWLCS: Oracle WebLogic Communication Services. Offers the TPCC service to Oracle applications and sets up the calls via SIP integration with the telephony network.

The set up task list Define WebLogic Communication Services Configuration delineates four tasks required for the correct configuration and implementation
of CTI. There is an optional task, separate from the set up task list, required for implementing Interaction logging.

Information about implementing CTI can be found in the Oracle Sales Cloud Administrator's Guide. Detailed information about configuring and maintaining WebLogic Communication Services is found in the Oracle WebLogic Communication Services Administrator's Guide

**Configure and Deploy WebLogic Server**

- Deploy WebLogic Communication Services: After the Oracle WebLogic communication server is deployed, this manual task activates the server.

  See: Oracle WebLogic Communication Services Administrator's Guide

**Integrate Communications Services**

- Integrate WebLogic Communication Services with Telephony Network: This manual task integrates communications within the telephony environment. OWLCS must be configured to interface with the specific characteristics of the telephony network.

  See: Managing Oracle WebLogic Communication Services for CTI Functionality

**Specify the Domain and Address**

- Register a URL for the telephony gateway or soft switch for SIP domain: This task defines the Server protocol, defaulted to http, the external server host address and external server port address. The Associated Modules section is not required for setup. You can also perform this as a manual task using Topology Manager to configure the address of the SIP Public Switched Telephone Network (PSTN) gateway or SIP soft switch serving the users within that domain. This address is needed by CTI to correctly form the SIP addresses required by WebLogic Communication Services.

  See the link to Configuring PSTN Gateway Address Using Topology Manager: Worked Example.

**Enable Click-to-Dial**

- After configuring the server and defining the SIP domain, perform the Enable Click-to-Dial task. This task sets the value of the profile option Enable Click-to-Dial to ‘Yes.’

**Call Logging via Interactions**

- To initiate the Interaction based logging for CTI, set the profile option Call Interaction Logging Enabled to ‘YES.’

**Configuring PSTN Gateway Address Using Topology Manager: Worked Example**

This example demonstrates how, during the implementation of the Register URL for the telephony gateway or soft switch for SIP domain task,
you must manually configure the PSTN gateway address by registering HzCTDPstnGatewayApp to a given environment using Oracle Fusion Topology Registration.

These steps configure the address of the SIP Public Switched Telephone Network (PSTN) gateway or SIP soft switch serving the users within that domain. This address is needed by Click-to-Dial to correctly form the SIP addresses required by WebLogic Communication Services.

For example: SIP:+1650-555-1212@pstn_gateway.oracle.com;user=phone where pstn_gateway.oracle.com is the SIP domain. The SIP domain can also be expressed in the format 10.1.1.1 (IP address).

Configuring PSTN using the Topology Manager

1. Sign in to Oracle Fusion Applications as a user that has application implementation consultant and WebLogic Services administration roles.

2. In Fusion Applications Setup and Maintenance, click Register Enterprise Applications from the regional area under Topology Registration.

3. On the Register Enterprise Applications page, click the plus icon to add an enterprise application. An Add Enterprise Application popup appears.

4. Enter the new application information: Click Search in the Enterprise Application list field. Enter HzCTDPstnGatewayApp in the name field and click Search. Click OK.

5. Enter the other fields in the Add Enterprise Application popup:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>HzCTDPstnGatewayApp</td>
</tr>
<tr>
<td>Server Protocol</td>
<td>SIP</td>
</tr>
<tr>
<td></td>
<td>This field is ignored by click-to-dial. Oracle WebLogic Communication Service (OWLCS) always uses the SIP protocol.</td>
</tr>
<tr>
<td>External Server Host</td>
<td>10.143.167.91 (Used as an example)</td>
</tr>
<tr>
<td></td>
<td>A host name can be used instead of an IP address.</td>
</tr>
<tr>
<td>External Server Port</td>
<td>0 (Used as an example)</td>
</tr>
<tr>
<td></td>
<td>This field is ignored by Click-to-Dial.</td>
</tr>
</tbody>
</table>

6. Click Save and Close
Define Applications Core Configuration

Define Lookups

**Lookups: Explained**

Lookups are lists of values in applications. You define a list of values as a lookup type consisting of a set of lookup codes, each code’s translated meaning, and optionally a tag. End users see the list of translated meanings as the available values for an object.

Lookups provide a means of validation and lists of values where valid values appear on a list with no duplicate values. For example, an application might store the values Y and N in a column in a table, but when displaying those values in the user interface, Yes or No (or their translated equivalents) should be available for end users to select. For example, the two lookup codes Y and N are defined in the REQUIRED_INDICATOR lookup type.

In another example, a lookup type for marital status has lookup codes for users to specify married, single, or available legal partnerships.

<table>
<thead>
<tr>
<th>Lookup Type</th>
<th>Lookup Code</th>
<th>Meaning</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAR_STATUS</td>
<td>M</td>
<td>Married</td>
<td></td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>Single</td>
<td></td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>Registered Partner</td>
<td>+NL</td>
</tr>
<tr>
<td></td>
<td>DP</td>
<td>Domestic Partner</td>
<td>-FR, AU</td>
</tr>
</tbody>
</table>

In this case, tags are used for localizing the codes. All legislations list Married and Single. Only the Dutch legislation lists Registered Partner. And all legislations except France and Australia also list Domestic Partner.

When managing lookups, you need to understand the following.

- Using lookups in applications
- Customization levels
- Accessing lookups
- Enabling lookups
• The three kinds of lookups: standard, common, and set enabled

Using Lookups in Applications

Use lookups to provide validation or a list of values for a user input field in a user interface.

An example of a lookup used for validation is a flexfield segment using a table-validated value set with values from a lookup type. An example of a lookup in a list of values is a profile option’s available values from which users select one to set the profile option. Invoice Approval Status gives the option of including payables invoices of different approval statuses in a report. The lookup code values include All so that users can report by all statuses: Approved, Resubmitted for approval, Pending or rejected, and Rejected.

Customization Level

The customization level of a lookup type determines whether the lookups in that lookup type can be edited. This applies data security to lookups.

Some lookup types are locked so no new codes and other changes can be added during implementation or later, as needed. Depending on the customization level of a lookup type, you may be able to change the codes or their meanings. Some lookups are designated as extensible, so new lookup codes can be created during implementation, but the meanings of predefined lookup codes cannot be modified. Some predefined lookup codes can be changed during implementation or later, as needed.

The customization levels are user, extensible, and system. The following table shows which lookup management tasks are allowed at each customization level.

<table>
<thead>
<tr>
<th>Allowed Task</th>
<th>User</th>
<th>Extensible</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deleting a lookup type</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Inserting new codes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Updating start date, end date, and enabled fields</td>
<td>Yes</td>
<td>Yes, only if the code is not predefined data</td>
<td>No</td>
</tr>
<tr>
<td>Deleting codes</td>
<td>Yes</td>
<td>Yes, only if the code is not predefined data</td>
<td>No</td>
</tr>
<tr>
<td>Updating tags</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Updating module</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Predefined data means LAST_UPDATED_BY = SEED_DATA_FROM_APPLICATION.

If a product depends on a lookup, the customization level should be system or extensible to prevent deletion.

Once the customization level is set for a lookup type, it cannot be modified. The customization level for lookup types created using the Define Lookups page is by default set at the User level.

Standard, Common, and Set-Enabled Lookups

The available kinds of lookups are as follows.
### 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.

For more information on the predefined lookups and lookup codes, see assets with the Lookup type in the Oracle Enterprise Repository for Oracle Fusion Applications (http://fusionappsoer.oracle.com).

### Managing a Standard Lookup: Example

Creating a new standard lookup involves creating or selecting a lookup type to which the lookup code belongs, and determining appropriate values for the lookup codes and their meanings.
Note
You can only create or edit the lookup codes for a particular lookup type if its customization level supports it.

Creating a Lookup Type Called COLORS

Your enterprise needs a list of values for status to be used on various objects such as processes or users. The lookups are colors, so the lookup type you create is COLORS.

<table>
<thead>
<tr>
<th>Lookup type parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup type name</td>
<td>COLORS</td>
</tr>
<tr>
<td>Meaning</td>
<td>Status</td>
</tr>
<tr>
<td>Description</td>
<td>Status by color</td>
</tr>
<tr>
<td>Module</td>
<td>Oracle Middleware Extensions for Applications</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Lookup Code</th>
<th>Meaning</th>
<th>Enabled</th>
<th>Display Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLUE</td>
<td>Urgent</td>
<td>No</td>
<td>4</td>
</tr>
<tr>
<td>RED</td>
<td>Stop</td>
<td>Yes</td>
<td>1</td>
</tr>
<tr>
<td>GREEN</td>
<td>Go</td>
<td>Yes</td>
<td>3</td>
</tr>
<tr>
<td>YELLOW</td>
<td>Caution</td>
<td>Yes</td>
<td>2</td>
</tr>
</tbody>
</table>

Understanding the Resulting Data Entry List of Values

Users need to respond to a process question by indicating whether to stop it, use caution, go ahead, or complete it urgently.

The list of values for the COLORS lookup type includes the meanings for the enabled codes.

<table>
<thead>
<tr>
<th>Displayed Value</th>
<th>Hidden ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop</td>
<td>RED</td>
</tr>
<tr>
<td>Caution</td>
<td>YELLOW</td>
</tr>
<tr>
<td>Go</td>
<td>GREEN</td>
</tr>
</tbody>
</table>

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.
Understanding the Transaction Table

When users enter one of the values from the list of values for the lookup type COLORS, the transaction table records the lookup code. In this example, the code is stored in the Status column.

<table>
<thead>
<tr>
<th>Transaction number</th>
<th>User name</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jane</td>
<td>RED</td>
</tr>
<tr>
<td>2</td>
<td>Bob</td>
<td>YELLOW</td>
</tr>
<tr>
<td>3</td>
<td>Alice</td>
<td>BLUE</td>
</tr>
</tbody>
</table>

The status for one user is BLUE because at the time they entered a value, BLUE was enabled. Disabling a lookup code does not affect transaction records in which that code is stored. Data querying and reporting have access to disabled lookup codes in transaction tables.

Managing Set-Enabled Lookups: Examples

Creating a new set-enabled lookup is similar to creating a standard lookup with the addition of specifying a reference data set determinant for the lookup codes.

Note

You can only create or edit the lookup codes for a particular lookup type if its customization level supports it.

The reference data set for a set-enabled lookup code is part of its foreign key. This is unlike other set-enabled entities.

Selecting a Reference Group for a Set-Enabled Lookup Type

By specifying a reference group for a set-enabled lookup type you indicate which reference data set assignments are available for its lookup codes. For example a COLORS lookup type might be set enabled for a Countries reference group that includes the US and EU reference data set assignments.

Selecting a Reference Data Set for a Set-Enabled Lookup

The reference data set determines which lookup code is included in the list of values. If a COLORS lookup type contains a RED, YELLOW, ORANGE, and GREEN lookup code, you can enable one RED lookup as coming from the US reference data set and another RED lookup as coming from the EU reference data set with divergent meanings.

<table>
<thead>
<tr>
<th>Reference Data Set</th>
<th>Lookup Code</th>
<th>Lookup Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>RED</td>
<td>Red</td>
</tr>
<tr>
<td>US</td>
<td>YELLOW</td>
<td>Yellow</td>
</tr>
<tr>
<td>US</td>
<td>GREEN</td>
<td>Green</td>
</tr>
<tr>
<td>EU</td>
<td>RED</td>
<td>Rouge</td>
</tr>
<tr>
<td>EU</td>
<td>ORANGE</td>
<td>Orange</td>
</tr>
</tbody>
</table>
In addition to divergent meanings for lookup codes based on associated reference data set, some lookup codes may be unique to one or another reference data set as the ORANGE lookup is to the EU reference data set in this example.

In another example, a lookup type called HOLD_REASON provides a list of reasons for applying a hold to a contract renewal. Reference data sets determine which codes are included in the hold reason list of values.

<table>
<thead>
<tr>
<th>Reference Data Set</th>
<th>Lookup Code</th>
<th>Lookup Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>SEC</td>
<td>SEC Compliance Review</td>
</tr>
<tr>
<td>US</td>
<td>DIR</td>
<td>Needs Director’s Approval</td>
</tr>
<tr>
<td>US</td>
<td>VP</td>
<td>Needs Vice President’s Approval</td>
</tr>
<tr>
<td>CHINA</td>
<td>CSRC</td>
<td>Pending China Securities Regulatory Commission Review</td>
</tr>
<tr>
<td>CHINA</td>
<td>PR</td>
<td>Needs President’s Approval</td>
</tr>
<tr>
<td>COMMON</td>
<td>REQUESTED</td>
<td>Customer Request</td>
</tr>
</tbody>
</table>

Using the Manage Set Assignments task, you have defined assignments that designate the China business unit to refer to the CHINA and the US business unit to refer to the US and all business units to refer to the COMMON set. When end users place a contract hold in the US business unit, only the three reason codes in US_SET are available. When placing a contract hold in the China business, only the two codes in China_SET are available.

FAQs for Define Lookups

How can I edit lookups?

You can edit the existing lookup codes of a lookup type or add new lookup codes on the Define Lookups pages, which you can access by starting in the Setup and Maintenance work area and searching for lookup tasks. You can edit the existing lookup codes of a lookup type, or add new lookup codes to a lookup type, if the customization level for the lookup type supports editing.

Why can't I see my lookup types?

Lookups are listed by lookup type. Typically lookup types are managed using tasks that handle a group of related lookups, such as Manage Geography Lookups. Each task gives you access only to certain lookup types. The generic tasks provide access to all lookups types of a kind, such as all common lookups using the Manage Common Lookups task.

If existing lookups are not available to the tasks of the Define Lookups activity, they may be validated for use in a lookup view that is not central to all applications or whose owning application has not been specified in a lookup view.

Lookups can only be managed in the Define Lookups tasks if the lookup’s view application is the standard lookups view, common lookups view, or set-enabled
Lookups view. Lookups defined in an application view can only be managed by following instructions provided by the owning application.

**Note**

A lookup type and its codes can only be defined in one lookup view.

---

**What's the difference between a lookup type and a value set?**

A lookup type consists of lookup codes that are the values in a static list of values. Lookup code validation is a one to one match.

A table-validated value set can consist of values that are validated through a SQL statement, which allows the list of values to be dynamic.

**Tip**

A table validated value set can be defined based on any table, including the lookups table. This allows a lookup type to be made into a table-validated value set that can be used in flexfields.

<table>
<thead>
<tr>
<th>Area of Difference</th>
<th>Lookup Type</th>
<th>Value Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of values</td>
<td>Static</td>
<td>Dynamic if Table validation type</td>
</tr>
<tr>
<td>Validation of values</td>
<td>One to one match of meaning to code included in a lookup view, or through the determinant of a reference data set</td>
<td>By format or inclusion in a table</td>
</tr>
<tr>
<td>Format type of values</td>
<td>char</td>
<td>varchar2, number, and so on</td>
</tr>
<tr>
<td>Length of value</td>
<td>Text string up to 30 characters</td>
<td>Any type of variable length from 1 to 4000</td>
</tr>
<tr>
<td>Duplication of values</td>
<td>Never. Values are unique.</td>
<td>Duplicate values allowed</td>
</tr>
<tr>
<td>Management</td>
<td>Managed by both administrators and end-users, except system lookups or predefined lookups at the system customization level, which cannot be modified.</td>
<td>Maintained by administrators, except some product flexfield codes, such as GL for Oracle Fusion General Ledger, which are maintained by end users</td>
</tr>
</tbody>
</table>

A lookup type cannot make use of a value from a value set.

Value sets can make use of standard, common, or set-enabled lookups.

Both lookup types and value sets are used to create lists of values from which users select values.

**What's a lookup tag used for?**

Tags on lookup codes allow you to add a label to your lookup codes.
Lookup tags are unvalidated and uninterpreted by lookups. A tag can be used to categorize lookups based on facilitating searches or guiding how a lookup should be used.

Document what the tag on a lookup represents and how to use it.

Manage Messages

Messages: Highlights

The message dictionary contains messages that tell users about business rule errors, such as missing or incorrect data, and how to resolve them, to warn users about the consequences of intended actions, and provide information in log files. These messages are defined for specific applications and modules, but a few are common messages that can be used in any application. All applications also use messages stored outside of the message dictionary.

The message dictionary is described in the Oracle Fusion Applications Developer's Guide.

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, such as confirmations and field validations, are managed either in the Oracle Application Development Framework or through message resource bundles used for translation.

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. To create or edit a message, navigate to the Manage Messages page in the Setup and Maintenance work area.

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_CNM 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_CNM_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 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.

**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.
Define Applications Core Configuration

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

Creation Mode

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

Customization

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

Multiple Tree Versions

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

Managing Tree Structures: Points to Consider

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

Creating and Editing Tree Structures

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

Note

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

Setting Status

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

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

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

In use and based on which one or more trees or tree versions are created.

Inactive

Not in use.

Tree Structure Audit Results: Explained

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

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

Running an Audit

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

Validation Details

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

<table>
<thead>
<tr>
<th>Validator</th>
<th>Description (what is checked)</th>
<th>Possible Cause for Validation Failure</th>
<th>Suggested Corrective Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restrict By Set ID</td>
<td>On the Manage Tree Structures: Specify Data Sources page, if the Set ID check box is selected to enable the Restrict Tree Node List of Values Based on option for a tree structure, each of its data source view objects must have a reference data set attribute. This validation does not take place when the check box is not selected.</td>
<td>Even when the check box is selected, one or more of its data source view objects does not contain a reference data set attribute.</td>
<td>If reference data set restriction is required for this tree structure, include a reference data set attribute on all data sources. Otherwise, deselect the check box.</td>
</tr>
</tbody>
</table>
| 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 FND_TREE_NODE_RF or a custom table. | • The specified table does not exist in the database.  
• The specified table does not contain the same columns as the FND_TREE_NODE_RF table. | Correct the row flattened table definition.                                                                 |

16-16 Oracle Product Value Chain Cloud Implementing Product Information Management
| Available Label Data Sources | On the Manage Tree Structures: Specify Data Sources page, if a labeling scheme is specified for the tree structure by selecting a list item from the **Labeling Scheme** list box, the label data source view object specified for each data source must be accessible, and the primary keys must be valid. This restriction does not apply when you select **None** from the **Labeling Scheme** list box. | • Any of the specified label data source view objects do not exist.  
• Any of the specified label data source view objects do not have primary keys.  
• When a label data source view object is initially defined, the database registers the primary keys for the view object. If the view object is later modified such that its primary keys no longer match the primary keys that were registered earlier, this validation fails.  
• Correct the specified label data source view object.  
• Correct the primary keys of the specified label data source view object.  
• Either correct the primary keys in the label data source view object to match the primary keys that were earlier registered in FND_TS_DATA_SOURCE, or correct the primary keys registered in that table to match the new view object definition. |
**Available Data Sources**

| Each data source view object specified for the tree structure must be accessible, and all its primary key attributes must be valid. | • Any of the specified data source view objects do not exist.  
• When a data source view object is initially defined, the database automatically registers the primary keys for the view object if the **Use non-defined primary key columns** check box on the Data Source dialog box is not selected. If the check box is selected, the database registers the primary keys specified explicitly by the user on the Add Data Source dialog box. If the registered primary keys contain any duplicates, this validation fails. | • Correct the specified data source view object.  
• Correct the duplicate column in the registered primary keys.  
• Correct the primary keys of the specified data source view object.  
• Correct any mismatch in data types. |
| --- | --- | --- |

• The **Use non-defined primary key columns** check box is selected in a data source, but the list of specified primary key columns does not match the primary keys defined in the corresponding data source view object.  
• Any common attribute that exists in both the data source view object and the tree node view object is not of the same data type in both view objects. |
### 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.
Oracle Product Value Chain Cloud Implementing Product Information Management

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

**Labeling Schemes**

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

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

**Restriction of Tree Node Values**

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

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

**Data Source Values and Parameters**

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

**Note**

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

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

- **Bound Value**: Captures any fixed value, which is used as part of the view criteria condition.
- **Variable**: Captures and binds a dynamic value that is being used by the data source view object. This value is used by the WHERE condition of the data flow.
- **View Criteria**: Captures the view criteria name, which is applied to the data source view object.
You can also specify which of the data source parameters are mandatory while creating or editing the tree structure.

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

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

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

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

**Row Flattening**

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

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

**Column Flattening**

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

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

**Column Flattened Entity Objects**

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

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

Manage Tree Labels

Tree Labels: Explained

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

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

<table>
<thead>
<tr>
<th>Labeling Scheme</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>Labels that are automatically assigned based on the data source to which the tree node belongs. A level label points to a specific data source. For example, in a tree that reflects the organizational hierarchy of an enterprise, all division nodes appear on one level and all department nodes on another.</td>
</tr>
<tr>
<td>Group</td>
<td>Labels that you can arbitrarily assign to tree nodes.</td>
</tr>
<tr>
<td>Depth</td>
<td>Labels that are automatically assigned based on the depth of the tree node within the tree. No manual assignment is performed.</td>
</tr>
</tbody>
</table>

Note

In an unbalanced hierarchy, a level may not be equal to depth.

Manage Trees and Tree Versions

Managing Trees and Tree Versions: Points to Consider

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

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

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

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

Creating and Editing Tree Versions

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

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

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

Tree Version Audit Results: Explained

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

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

Running an Audit

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

Validation Details

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

<table>
<thead>
<tr>
<th>Validator</th>
<th>Description (what is checked)</th>
<th>Possible Cause for Validation Failure</th>
<th>Suggested Corrective Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effective Date</td>
<td>The effective start and end dates of the tree version must be valid.</td>
<td>The effective end date is set to a value that is not greater than the effective start date.</td>
<td>Modify the effective start and end dates such that the effective start date is earlier than the effective end date.</td>
</tr>
<tr>
<td>Root Node</td>
<td>On the Manage Tree Structures: Specify Data Sources page, if the <strong>Allow Multiple Root Nodes</strong> check box for the <strong>Restrict Tree Node List of Values Based on</strong> option is not selected, and if the tree structure is not empty, the tree version must contain exactly one root node. This validation does not take place if the check box is selected.</td>
<td>Even if the check box is deselected, the tree version has multiple root nodes.</td>
<td>Modify the tree version such that there is exactly one root node.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Data Source Max Depth</td>
<td>For each data source in the tree structure, on the Data Source dialog box, if the data source is depth-limited, the data in the tree version must adhere to the specified depth limit. This validation does not apply to data sources for which the <strong>Maximum Depth</strong> field is set to <strong>Unlimited</strong>.</td>
<td>The tree version has data at a depth greater than the specified depth limit on one or more data sources.</td>
<td>Modify the tree version such that all nodes are at a depth that complies with the data source depth limit.</td>
</tr>
<tr>
<td>Duplicate Node</td>
<td>On the Data Source dialog box, if the <strong>Allow Duplicates</strong> check box is not selected, the tree version should not contain more than one node with the same primary key from the data source. If the check box is selected, duplicate nodes are permitted.</td>
<td>Even when the check box is deselected, the tree version contains duplicate nodes.</td>
<td>Remove any duplicate nodes from the tree version.</td>
</tr>
<tr>
<td>Available Node</td>
<td>All nodes in the tree version should be valid and available in the underlying data source.</td>
<td>• A node in the tree version does not exist in the data source. Deleting data items from the data source without removing the corresponding nodes from the tree version can result in orphaned nodes in the tree version. For example, if you added node A into your tree version, and subsequently deleted node A from the data source without removing it from the tree version, the validation fails. • The tree version contains a tree reference node, which references another tree version that does not exist.</td>
<td>Remove any orphaned nodes from the tree version. Update tree reference nodes so that they reference existing tree versions.</td>
</tr>
<tr>
<td>Node Relationship</td>
<td>All nodes must adhere to the relationships mandated by the data sources registered in the tree structure.</td>
<td>The tree structure has data sources arranged in a parent-child relationship, but the nodes in the tree do not adhere to the same parent-child relationship. For example, if the tree structure has a Project data source with a Task data source as its child, Task nodes should always be under Project nodes in the tree version. This validation fails if there are instances where a Project node is added as the child of a Task node.</td>
<td>Modify the tree version such that the nodes adhere to the same parent-child relationships as the data sources.</td>
</tr>
<tr>
<td>SetID Restricted Node</td>
<td>On the Manage Tree Structures: Specify Data sources page, if the Set ID check box is selected to enable the <strong>Restrict Tree Node List of Values Based on</strong> option for each tree node, the underlying node in the data source must belong to the same reference data set as the tree itself. This restriction does not apply when the check box is not selected.</td>
<td>Even when the check box is selected, the tree version has nodes whose data source values belong to a different reference data set than the tree.</td>
<td>Modify the tree version such that all nodes in the tree have data sources with reference data set matching that of the tree.</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Label Enabled Node</td>
<td>On the Manage Tree Structures: Specify Data Sources page, if a labeling scheme is specified for the tree structure by selecting a list item from the <strong>Labeling Scheme</strong> list box, all nodes should have labels. This restriction does not apply when you select <strong>None</strong> from the <strong>Labeling Scheme</strong> list box.</td>
<td>The tree structure has a labeling scheme but the tree version has nodes without labels.</td>
<td>Assign a label to any node that does not have a label.</td>
</tr>
<tr>
<td><strong>Date Restricted Node</strong></td>
<td>On the Manage Tree Structures: Specify Data Sources page, if the <strong>Date Range</strong> check box is selected to enable the <strong>Restrict Tree Node List of Values Based on</strong> option for a tree structure, each node in the underlying data source must have an effective date range same as the effective date range of the tree version. This restriction does not apply if the check box is not selected.</td>
<td>Even when the check box is selected, there are data source nodes that have a date range beyond the tree version’s effective date range. For example, if the tree version is effective from Jan-01-2012 to Dec-31-2012, all nodes in the tree version must be effective from Jan-01-2012 to Dec-31-2012 at a minimum. It is acceptable for the nodes to be effective for a date range that extends partly beyond the tree version’s effective date range (for example, the node data source value is effective from Dec-01-2011 to Mar-31-2013). It is not acceptable if the nodes are effective for none or only a part of the tree version’s effective date range (for example, the node data source value are effective only from Jan-01-2012 to June-30-2012).</td>
<td>Ensure that all nodes in the tree version have effective date range for the effective date range for the tree version.</td>
</tr>
<tr>
<td><strong>Multiple Active Tree Version</strong></td>
<td>On the Manage Tree Structures: Specify Definition page, if the <strong>Allow Multiple Active Tree Versions</strong> check box is not selected for the tree structure, there should not be more than one active tree version under a tree at any time. This restriction does not apply if the check box is selected.</td>
<td>Even when the check box is not selected, there is more than one active tree version in the tree for the same date range.</td>
<td>Set no more than one tree version to Active within the same date range and set the others to inactive or draft status.</td>
</tr>
<tr>
<td><strong>Range Based Node</strong></td>
<td>On the Data Source dialog box, if the <strong>Allow Range Children</strong> check box is not selected, range-based nodes are not permitted from that data source. This restriction does not apply if the check box is selected.</td>
<td>Even when the check box is not selected, there are range-based nodes from a data source.</td>
<td>Ensure that any range nodes in your tree version are from a data source that allows range children.</td>
</tr>
</tbody>
</table>
### Terminal Node

On the Data Source dialog box, if the **Allow Use as Leaves** check box is not selected, values from that data source cannot be added as leaves (terminal nodes) to the tree version. This restriction does not apply if the check box is selected.

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 **Use All Values** option is selected to set the **Usage Limit** for the data source, every value in the data source must appear as a node in the tree. This restriction does not apply if **None** option is selected.

Even if the **Use All Values** 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
Define Applications Core Configuration

16-29

Table or in an entity that has been established by tree management as the default storage mechanism. However, since all data in Oracle Fusion Applications usually have a storage home, only user-created data needs to be stored in an entity.

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

Managing Tree Nodes

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

Node Levels

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

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

Node Types

A tree node has the following node types.

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

Define Profile Options

Profile Options: Explained

Profile options manage configuration data centrally and influence the behavior of applications. Profile options serve as permanent user preferences and
application configuration parameters. You configure profile options with settings for specific contexts or groups of users. Users customize how their user interfaces look and behave by changing the values of available profile options.

Profile options store the following kinds of information.

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

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

<table>
<thead>
<tr>
<th>Profile Option Level</th>
<th>Value of the Profile Option Level</th>
<th>Profile Option Value</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Manager1</td>
<td>UK pound sterling</td>
<td>Access to site and all products shows UK pounds sterling in effect</td>
</tr>
</tbody>
</table>
Define Applications Core Configuration

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

Context such as user session or accessed product determines which profile option value is associated with the profile option name. In the example, if manager1 does not set a profile option value for this profile option, access to Financials for EMEA shows currency in Euros; and access to other products shows currency in UK pounds sterling.

**Profile Option Categories**

Categories group profile options based on their functional area. Profile option categories facilitate searching and defining data security.

For example, in Oracle Fusion Receivables, the Transactions profile option category groups profile options related to setting how Receivables transactions are to be processed, such as Require Adjustment Reason.

A profile option can be in more than one category.

**Profile Option Hierarchies and Levels**

Application developers specify at which hierarchy level a profile option is enabled. The predefined profile option hierarchy levels are site, product, and user.

The hierarchy levels specified in the profile option definition determine the context in which a profile option value may be set. If the profile option value at a particular level is updatable, an administrator can update the profile option value for that context.

**Note**

Profile options should only be enabled for context levels that are appropriate for that profile option. For example, a profile option indicating a global configuration setting should not be enabled at the user level, if users cannot choose a different value for that setting.

For security, one level in the hierarchy is designated as a user level. A profile option may be enabled at any or all hierarchy levels. When enabled at all levels, the predefined ordering of profile option hierarchy levels gives precedence to the values that are set at the user level over values set at the product and site levels, and precedence to values set at the product level to values set at the site level. If there is no value for the current user, then the product value applies. If there is no value for the user or product, then the site value applies.

The table shows the predefined profile option hierarchy and ordering.
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.

For more information on the predefined profile options, see assets with the Profile Option type in the Oracle Enterprise Repository for Oracle Fusion Applications (http://fusionappsoer.oracle.com).

**Planning Profile Options: Points to Consider**

Plan profile options before defining and configuring them.

The following aspects assist you in better planning how to manage profile options.

- Profile option tasks
- Before creating a profile option
- Profile options data model

**Profile Option Tasks**

Users may be able to set their own profile options, depending on settings in the profile option definition. However, not all profile options are visible to end users, and some profile options, while visible, may not be updated by end users.
The following table lists tasks and considerations relevant to planning profile options.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Role</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning, creating, and editing a new profile option</td>
<td>Applications developer</td>
<td>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.</td>
</tr>
<tr>
<td>Configure values in an existing profile option</td>
<td>Applications developer, application administrator, and implementation consultant</td>
<td>Manage the values for existing profile options.</td>
</tr>
<tr>
<td>Create and edit profile option categories</td>
<td>Applications developer, application administrator, and implementation consultant</td>
<td>Manage categories for organizing existing profile options.</td>
</tr>
</tbody>
</table>

**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.

```sql
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_LANGUAGE_VL`.

<table>
<thead>
<tr>
<th>Display Value</th>
<th>Hidden Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>American English</td>
<td>US</td>
</tr>
<tr>
<td>French</td>
<td>F</td>
</tr>
<tr>
<td>Spanish</td>
<td>E</td>
</tr>
</tbody>
</table>

Hidden values must be `varchar2(2000)`.

Profile options generally provide configuration values within a particular context. Though you can create a profile option to be global, think of global values as default values to avoid storing inappropriate configuration information as profile option values. Create global profile options that have corresponding contextual levels.

**Managing Profile Option Categories: Points to Consider**

Use profile option categories to group profile options.

**Organizing Profile Options in Categories**

As a guideline, group profile options in a single category if the profile options affect the same feature, or if an administrator would likely want to see the profile options in the results of a single search.

Application developers are responsible for the initial groupings and then administrators can make changes based on their specific needs. Administrators
can categorize profile options and then easily search on profile options by category.

---

**Tip**

Define profile option categories first and assign new profile options to existing categories rather than defining profile options first and then defining categories to categorize them.

---

**Adding New Profile Option Categories**

You can add new categories or add profiles to an existing category.

You can create a profile option category by duplicating an existing category and editing it for a new grouping of profile options. You can add multiple profile options to a category. A profile option can exist in multiple categories.

**Profile Option Order in a Category**

Specify a profile option sequence to determine the order of profile options when queried by profile option category.

---

**Viewing and Editing Profile Option Values: Points to Consider**

A profile option value consists of the value and the context or level where the value is set. You specify the context with a pairing of the profile option value’s level and level value, such as the product level and the level value GL for Oracle Fusion General Ledger. Adding or modifying profile option values can include deciding which valid values are enabled or updatable at which level.

The SQL validation of the profile option definition determines what valid profile option values are available. In the absence of validation, any value is valid.

**Profile Option Levels and User Session Context**

Site level profile option values affect the way all applications run for a given implementation. Product level profile option values affect the way applications owned by a particular product code behave. For example, a product may use profile options set at the product level to determine how regions provided by a common module such as those available from Oracle Fusion Trading Community Model or Customer Relationship Management (CRM) display in a particular work area or dashboard. User level profile option values affect the way applications run for a specific application user.

Whichever profile option value is most specific to a user session, that is the value at which the profile option is set for the user session.

For example, the predefined FND_LANGUAGE profile option sets the default language. In addition to a site level value, you can define a value for various product or user levels.
Values at the site level take effect for any user unless overridden by a different value set at the more specific levels of product and user. Product level profile option values affect the way applications owned by a particular product code behave. In addition to user level profile option values in applications, selections may be available in the user preferences workspace.

The following table demonstrates the FND_LANGUAGE profile option settings that would apply to specific users, based on the example above. For example, the user Hima is using the CRM Application Composer product, in the InFusion site. The example above shows that this profile option is set to Hindi at the user level for Hima. Because user is the highest applicable level for Hima, the applicable profile option value is Hindi for Hima.

<table>
<thead>
<tr>
<th>Site</th>
<th>Product</th>
<th>User</th>
<th>Highest Available Level</th>
<th>Active Profile Option Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>InFusion</td>
<td>CRM Application Composer</td>
<td>Hima</td>
<td>User</td>
<td>Hindi</td>
</tr>
<tr>
<td>Acme</td>
<td>Payables</td>
<td>Application Administrator</td>
<td>User</td>
<td>American English</td>
</tr>
<tr>
<td>InFusion</td>
<td>Customer Center</td>
<td>Guillaume</td>
<td>Product</td>
<td>French</td>
</tr>
<tr>
<td>InFusion</td>
<td>Payables</td>
<td>Implementation Consultant</td>
<td>Site</td>
<td>American English</td>
</tr>
<tr>
<td>Acme</td>
<td>Payables</td>
<td>Implementation Consultant</td>
<td>none</td>
<td>no value</td>
</tr>
</tbody>
</table>

**Note**

More than one site level value is relevant in an enterprise with multiple tenants using a single instance of Oracle Fusion Applications.

**Effect of Changes to Profile Option Values**

Any change you make to a user level profile option has an immediate effect on the way applications run for that session. When you sign in again, changes made to your user level profile options in a previous session are still in effect. When you change profile option value at the product level and no user level values are set, you see the update immediately, but other users may not see the changed value until signing out and back in. When you change a profile option value and the new value affects other users, the change takes effect only when users sign in the next time.

Changes to site level profile options take effect for any user session that is started after the setting has been changed. Changes to site or user level profile options
do not affect any user sessions that are already in progress when the change is made.

Changes to site or user level profile options take effect for any C or PL/SQL processes, such as scheduled jobs, that are launched after the setting has been changed. Profile option changes do not affect C or PL/SQL processes that are already running.

Define Flexfields

Flexfields: Overview

A flexfield is an extensible set of placeholder fields in application pages that are associated with a business object. Each segment of the flexfield corresponds to a single application field, such as a segment of a key identifying a particular purchase, or the components of a student's contact information, or the features of a product in inventory.

Using descriptive and extensible flexfields, you can extend business objects to capture data that wouldn’t otherwise be tracked by the application. If you need to add custom fields to a business object to meet your enterprise-specific requirements, configure the flexfield to have one segment for each needed field.

Using key flexfields, you can configure intelligent key codes comprised of meaningful parts according to your business practices. You configure the key flexfield to have one segment for each part that makes up your key code.

Flexfields let you meet enterprise requirements without changing the data model. Different data can be captured on the same database table. Each segment captures a single atomic value, has a name, and maps to a pre-reserved column in the application database.

You can use a flexfield to extend a business object if it has been registered for use on that object. Application developers create a flexfield and register it so that it is available for configuration. Administrators and implementation consultants set up or configure segments and other properties of the available flexfields. End users see flexfield segments as fields or 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.

The following aspects provide an overview of flexfields:

- Accessing flexfields and flexfield management tasks
- Types of flexfields
- Flexfield segments
- Value sets
- Structure and context
- Deployment
• Run time appearance

**Accessing Flexfields and Flexfield Management Tasks**

You can view flexfields on a page where they occur using the Highlight Flexfields feature. You can access flexfield management tasks directly from a highlighted flexfield, through product-specific flexfield management tasks, or by starting in the Setup and Maintenance Overview page which is available from the Navigator or the Administration menu.

For lists of flexfields, see assets with the Flexfield: Descriptive, Flexfield: Extensible, or Flexfield: Key type in Oracle Enterprise Repository for Oracle Fusion Applications (http://fusionappsoer.oracle.com).

**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 lets you collect custom invoice details fields on an invoices page. You can implement these fields, which are descriptive flexfield segments, as context-sensitive so they appear only when needed on a row-by-row basis when specific contextual information is met. 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 custom fields needed beyond the predefined fields.

**Segments**

Each field that you configure using flexfields is a flexfield segment. 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 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 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 run time 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.

**Run time 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 flexfields on the run time page. Flexfields in highlight mode display an **Information** icon button to
access details about the flexfield, an Edit icon button to manage the flexfield, and an Add Segment icon button to add flexfield segments.

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.

**Configuring Flexfields: Overview**

Configuring a flexfield ranges from identifying the need for extending a business object with custom attributes to integrating the custom attributes into the deployment. In the case of key flexfields, configuring the flexfield involves identifying value set assignments and determining segment structures.

**Overall Process for Configuring Custom Attributes**

For descriptive and extensible flexfields, the overall configuration process involves the following:

1. Use the Highlight Flexfields feature from the Administration menu to find flexfields on pages associated with business objects.
2. Plan the flexfield configuration.
3. Plan flexfield validation.
4. Define the attributes by configuring the flexfield segments.
   a. Use the Manage Extensible Flexfields or Manage Descriptive Flexfields tasks, or use the Configure icon button directly on the page where the flexfield is highlighted. For simple configurations, use the Add Segment, Add Context Value, and Edit Segment icon buttons directly on the page where the flexfield is highlighted.
   b. Optionally, validate the flexfield configuration.
   c. Optionally, deploy the flexfield to a sandbox for initial testing.
5. Deploy the flexfield to the mainline to display the custom attributes on the application pages and to make them available for integration with other tools such as Oracle Business Intelligence.
6. Perform the necessary steps to integrate the custom attributes into the technology stack.

A simple configuration is limited to such actions as adding a format-only field or adding a field with a basic list of values.

**Overall Process for Configuring Custom Keys**

Using key flexfields, you can configure intelligent key codes comprised of meaningful parts according to your business practices. You configure the key flexfield to have one segment for each part that makes up your key code.
For key flexfields, the overall configuration process involves the following:

1. Use the Highlight Flexfields feature from the Administration menu to find flexfields on pages associated with business objects.

2. Plan the flexfield configuration.

3. Plan the flexfield validation.

4. Define the value sets before configuring the key flexfield segments by going to the Manage Value Sets task.

5. Define the key flexfield structures and their segments, and define structure instances for each structure.
   a. Use the Manage Key Flexfields task or the **Configure** icon button directly on the page where the flexfield is highlighted.
   b. Optionally, validate the flexfield configuration.
   c. Optionally, deploy the flexfield to a sandbox for initial testing.

6. Deploy the flexfield to the mainline to display it on the application pages and to make it available for integration with other tools such as Oracle Business Intelligence.

7. Perform the necessary steps to integrate the flexfield into the technology stack.

**Flexfields at Run Time: Explained**

Many business objects in Oracle Fusion applications have an associated descriptive or extensible flexfield with which you can create custom attributes for the business object. Some business objects have an associated key flexfield for configuring flexible multiple part keys.

The following aspects are important in understanding flexfields at run time:

- Finding flexfields on a page
- Why no flexfields are on a page

**Finding Flexfields on a Page**

At run time, the custom attributes you define as extensible and descriptive flexfield segments appear in the application page just like any other attribute. Key flexfields typically appear in the application page as a field with a key flexfield icon, where the field’s value is actually a collection of segments. In some pages, one or more key flexfield segments may be displayed in the page like any other attribute. Thus, when viewing the page in standard mode, in many cases you may not be able to discern which fields are flexfield segments, or whether flexfields are available to configure on the page.

Use the Highlight Flexfields feature to render the page in a special mode that lets you view:

- Where, if any, flexfields are available on your page
• Which, if any, of the fields on your page are flexfield segments rather than out-of-the-box fields

To obtain information about the flexfields on a page, open the page and choose Highlight Flexfields from the Administration menu. Hover over the Information icon button next to the highlighted fields to display information about the flexfield. Choose Unhighlight Flexfields from the Administration menu when you no longer want to see the highlighted flexfields.

When you click the Configure Flexfield icon button for a highlighted flexfield, the applicable Manage Flexfields task is displayed for that flexfield. For simple configurations, you can click the Add Context Value icon button to add a context value, or click the Add Segment or Edit Segment icon buttons to add or edit a global segment or a context-sensitive segment that doesn’t require advanced configuration.

Note
Not all flexfields are available for creating custom attributes. Consult the product-specific documentation in Oracle Fusion Applications Help to verify whether there are any restrictions on using the flexfield.

Why No Flexfields Are on a Page

For a flexfield to be available in the page, it must be registered by developers. If a flexfield is available, you may configure segments. The segments appear on the page only after you have successfully deployed the flexfield. For information about registering flexfields, see the Oracle Fusion Applications Developer’s Guide. Some business objects haven’t been designed to support flexfields. For information about how to enable business objects with flexfield capability, see Getting Started with Flexfields in the Oracle Fusion Applications Developer’s Guide.

Note
The following Oracle Sales Cloud applications don’t support flexfields:

• Sales
• Marketing
• Customer Center
• Trading Community Architecture
• Order Capture

To add custom attributes to these applications, use Application Composer. For more information, see the “Editing an Object: Explained” section in Oracle Sales Cloud: Extending Sales.

Customizing Flexfields Using Page Composer: Explained

Using Page Composer, you can create customizations to flexfields that are specific to a page.
In Page Composer, to customize:

- Extensible flexfields, open the page in Source view, and look for a region that is bound to an EffContextsPageContainer task flow. This is the container for the extensible flexfield attributes and contexts. To view the flexfield code and identifying information, open the properties panel for the region. To customize any component within the region, select the desired tag and click Edit.

- Descriptive flexfields, open the page in Source view, and look for <descriptiveFlexfield> elements. Open the properties panel for the element to view the flexfield code and identifying information. Within the properties panel, you may customize properties for the global and context-sensitive segments or re-order the segments on the page.

**Accessing Flexfield Management Tasks: Procedures**

You can configure and manage flexfields by highlighting them on an application page and using the available on-screen configuration tools. Alternatively, you can access product-specific flexfield tasks or global flexfield management tasks.

**Accessing Flexfield Management Tasks through the Run time Page**

You can identify flexfields on the run time application page where they are implemented.

1. Navigate to an application page.
2. Choose Highlight Flexfields from the Administration menu in the global area of Oracle Fusion Applications.
3. View the available flexfields highlighted on the page. If any of the fields on the page are custom fields configured as part of a flexfield, they also appear highlighted.
4. To edit a flexfield, use the:
   - Configure Flexfield icon button to access the flexfield management task pages for extensive configuration to the flexfield and its segments.
   - Add Segment icon button to access the subwindow for adding segments with limited configuration to descriptive flexfields.
   - Edit Segment icon button to access the subwindow for limited configuration changes to descriptive flexfield segments.

**Accessing Flexfield Management Tasks through Setup and Maintenance**

Manage flexfields using tasks you access by starting in the Setup and Maintenance Overview page which is available from the Navigator or the Administration menu.

To access tasks for configuring flexfields and value sets, you must be provisioned with roles that entitle you to access the Define Flexfields task list or tasks for managing product-specific flexfields. Contact your security administrator for
Define Applications Core Configuration

For information about product-specific flexfield tasks, such as Manage Purchasing Descriptive Flexfields, consult the product-specific documentation in Oracle Fusion Applications Help.

To access the flexfield management tasks and search for existing flexfields, perform the following steps:

1. Choose **Setup and Maintenance** from the **Administration** menu in the global area of Oracle Fusion Applications.
2. Search for Define Flexfields in the All Tasks tab.

**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.
  - Application Flexfield Value Set to find all tasks related to value sets.
- 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.
  - Value set across all Oracle Fusion Applications products, search for the Define Flexfields task list and access the Manage Value Sets task.

**Restriction**

If you are configuring key flexfields, search for and access the Manage Value Sets task to set up value sets before accessing the Manage Key Flexfields task.

3. Expand the task list to view the included tasks.
4. Click the **Task** icon button to open the manage flexfield pages.
5. Search for all or specific flexfields.
6. In the search results, select the flexfield.
7. Use the Edit action to open pages for viewing and configuring the flexfield. Access to managing value sets is available within the tasks for managing descriptive and extensible flexfields.

**Note**

Access to managing value sets is:
- Available within the tasks for managing descriptive and extensible flexfields.
- Not available within the tasks for managing key flexfields. Therefore, configure value sets prior to configuring your key flexfield.
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.

For example, an airline manufacturer might require very specific attributes for their orders that aren’t provided by the out-of-the-box implementation of an order. Because a flexfield exists for the order business object, you can use it to create and configure the desired attribute.

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 ADF business component (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.

The following aspects are important in understanding how flexfields and Oracle Fusion Applications architecture work together:

- Integration
- Deployment
- Import and Export
- Run time
- Patching

**Integration**

The attributes that you add by configuring flexfields are available throughout the Oracle Fusion Middleware technology stack, allowing the flexfields to be used in user interface pages, incorporated into the service-oriented architecture (SOA) infrastructure, and integrated with Oracle Business Intelligence. You identify flexfield segments for integration by the segment's Application Programming Interface (API) name.

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 run time 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.

**Run time**

For a flexfield to reflect the latest flexfield definition at run time 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.

**Patching**

Flexfield configurations are preserved during patching and upgrading.

**Flexfields and Value Sets: Highlights**

Before you use flexfields to create custom attributes, you should be familiar with the Oracle Fusion application architecture that enables customization, customization layers, and the customization lifecycle.

In addition to the extensive information in the Oracle Fusion Applications Help about configuring flexfields that are already available for configuration, consider the resources below for adding flexfields to business components and alternatives to flexfields where flexfields cannot be enabled.

To assess the flexfields available in a deployment of Oracle Fusion Applications, see assets of type: flexfield in the Oracle Enterprise Repository at http://fusionappsoer.oracle.com.


**Restriction**

Don’t use Oracle JDeveloper to customize flexfields.

**Before Configuring Flexfields**

You can add custom attributes to a business object using a flexfield, if a flexfield has been registered for that object by developers.

- For information about registering flexfields to business objects, refer to the Oracle Fusion Applications Developer's Guide.
  
  See: Getting Started with Flexfields

  - The user interface page for a business object that a developer extends to support a flexfield must be enabled to display the custom attributes defined by the flexfield.

  See: Adding Descriptive Flexfield UI Components to a Page

  See: Employing an Extensible Flexfield on a User Interface Page

- For Sales, Marketing, Customer Center, Trading Community Architecture, and Order Capture applications, use Application Composer to add custom attributes instead of using descriptive and extensible flexfields. For more information, refer to Oracle Sales Cloud: Extending Sales.

  See: Application Composer: Introduction

- For information about displaying translated values of a table-validated value set from the value column for the run time session's locale, refer to the Oracle Fusion Applications Developer's Guide.
See: Using Multi-Language Support Features

Security

- For an understanding of data security when considering the consequences of applying data security to value sets, refer to the Oracle Fusion Applications Security Guide.

See: Data Security

Deploying Flexfields

- To examine the artifacts of a deployed flexfield configuration that you exported using the exportMetadata WLST command, refer to the Oracle Fusion Applications Extensibility Guide.

See: Exporting Customizations

- For information about synchronizing the updated XML schema definition (XSD) files in the metadata services (MDS) repositories for each service-oriented architecture (SOA) application, refer to the Oracle Fusion Applications Extensibility Guide.

See: Customizing SOA Composite Applications

- For information about incorporating a deployed flexfield into the technology stack, such as customizing the pages, integrating with Oracle Business Intelligence, or integrating into Web Services and service-oriented architecture SOA infrastructure, refer to the Oracle Fusion Applications Concepts Guide.

See: Oracle Fusion Middleware Components

- Oracle ADF services used by SOA composites expose the Web Services Description Language (WSDL) schemas where deployed flexfields are stored.

See: Oracle Fusion Middleware Developer’s Guide for Oracle SOA Suite

Oracle Business Intelligence

- For more information about importing and propagating your flexfield changes, refer to the Oracle Fusion Applications Extensibility Guide.

See: Customizing the Oracle BI Repository (RPD)

- For information about importing business intelligence-enabled flexfield changes into the Oracle Business Intelligence repository, refer to the Oracle Transactional Business Intelligence Administrator’s Guide.

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 let administrators configure and deploy flexfields. If you deploy a flexfield to a sandbox and decide to apply the configuration to the mainline, select the flexfield in the Manage Flexfields tasks of the Define Flexfields activity and deploy the flexfield in the mainline so that it is available to users.

Consider the following aspects of managing flexfields:

- Registering flexfields
- Planning flexfields
- Configuring flexfields
- Enabling a flexfields segment for business intelligence
- Deploying flexfields
- Optionally changing a flexfield segment's appearance in a user interface page
- Identifying flexfields on a run time page and troubleshooting
Registering Flexfields

Application development registers flexfields so they are available to administrators and implementation consultants for configuration.

As part of registering a flexfield, application development reserves columns of entity tables for use in flexfields so an enterprise can capture segments to meet their business needs. Many flexfields are registered 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

Before you begin planning flexfields, determine what type is appropriate to your needs, and which business objects are available for customizing 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.

Carefully plan flexfields before configuring them. Before configuring new segments for your flexfields, be sure to plan their implementation carefully.

If you have determined that a business object supports flexfields, and those flexfields have been registered, you can begin planning how to configure the flexfield for your needs. Note the code name of the flexfield you intend to configure so you can find it easily in the Define Flexfield activity.

In some cases you can customize how the flexfield appears on the page.

See Oracle Fusion Applications Help for specific products to determine any restrictions on using product-specific flexfields.

Configuring Flexfields

Administrators or implementers configure flexfields so they meet the needs of the enterprise. Some flexfields require configuration to make an application operate correctly.

You can configure flexfields using the following methods:

- Go to the manage flexfield tasks in the Setup and Maintenance work area.
- Use the Highlight Flexfields command in the Administration menu while viewing a run time page.
- Use the Configure Flexfield icon button to manage a flexfield, such as change a segment’s sequence number, or configure a flexfield segment’s business intelligence label.
- Use the Add Segment icon button to add descriptive flexfield segments and context values, or extensible flexfield segments.

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 identifying information for each segment
• Specifying the display properties such as 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
You can create value sets while creating descriptive and extensible flexfield segments. However, define value sets before configuring key flexfield segments that use them, because you assign existing value sets while configuring key flexfield segments.

When creating table-validated, independent, dependent, or subset value sets while creating descriptive and extensible flexfield segments, you can optionally specify to display the description of the selected value to the right of the segment at run time.

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 based on the segments’ sequence numbers. You cannot enter a number for one segment that is already in use for a different segment.

Tip
Consider numbering the segments in multiples, such as 4, 5, or 10, to make it easy to insert new attributes.

A flexfield column is assigned to a new segment automatically, but you can change the assignment before saving the segment. If you need to set a specific column assignment for a segment, create that segment first to ensure that the intended column isn’t automatically assigned to a different segment.

Enabling a Flexfield Segment for Business Intelligence

You can enable 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.

For extensible flexfield segments, you can’t assign labels and use equalization to prevent duplication.

Deploying Flexfields

Once you have configured a flexfield, you must deploy it to make the latest definition available to run time users.

In the Define Flexfields tasks, you can deploy a flexfield using either of the following commands:
• The Deploy Flexfield command to deploy a flexfield to mainline. This is for general use in a test or production environment.
• The Deploy to Sandbox command to deploy a flexfield to sandbox. This is to confirm that the flexfield is correctly configured before deploying it to the mainline.

When using the Add Segment and Edit Segment tools for descriptive flexfields in Highlight Flexfields mode, you can use the Save and Deploy command to save your changes and deploy the flexfield to mainline.

Once deployed, the deployment status indicates the state of the currently configured flexfield relative to the last deployed definition.

**Optionally Changing a Flexfield Segment Appearance**

The flexfield attributes that 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.

After a flexfield has been deployed to a mainline metadata services (MDS) repository so that it appears on application pages, you can customize it on a per-page basis using Page Composer. For example, you can hide a segment, change its prompt or other properties, or reorder the custom global attributes so that they are interspersed with the core attributes in the same parent layout.

You can only customize the appearance of descriptive and extensible flexfield segments in the UI page using Page Composer once the flexfield is deployed to the mainline.

If the Oracle Fusion applications are running in different locales, you can provide different translations for translatable text, such as prompts and descriptions. Enter translations by signing in using the locale that requires the translated text. You do this by selecting Settings and Actions - Personalization - Set Preferences in the global area and changing the text to the translated text for that locale.

**Identifying Flexfields on a Run time Page and Troubleshooting**

The Highlight Flexfields command in the Administration menu of the Setup and Maintenance work area identifies the location of flexfields on the run time page by displaying an Information icon button for accessing details about each flexfield.

Even if a descriptive or extensible flexfield hasn't yet been deployed and no segments appear on the run time page in normal view, the flexfield appears in the Highlight Flexfield view for that page. In the case of descriptive flexfields, the segments as of the last deployment appear. Highlight Flexfields accesses the current flexfield metadata definition.

Use the highlighted flexfield's Edit icon button to manage flexfields directly. Alternatively, note a highlighted flexfield's name to search for it in the tasks for managing flexfields.

To examine a flexfield's configuration, export the deployed artifacts using the exportMetadata WLST.

For more information on creating flexfields and adding them to a UI page, see the Oracle Fusion Applications Developer's Guide.
For more information about customizing flexfield segment appearance with Oracle Composer, see guidance on customizing existing pages in the Oracle Fusion Applications Extensibility Guide.

**Flexfield Segment Properties: Explained**

Independent of the value set assigned to a segment, segments may have properties that affect how they are displayed and how they behave. The following aspects are important in understanding:

- Display properties
- Properties related to segment values
- Properties related to search
- Range validation segments
- Rule validation of segment values
- Naming conventions

**Display Properties**

The following table summarizes display properties.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>Whether the segment can be used.</td>
</tr>
<tr>
<td>Sequence</td>
<td>The order the segment appears in relation to the other configured segments.</td>
</tr>
<tr>
<td>Prompt</td>
<td>The string to be used for the segment’s label in the user interface.</td>
</tr>
<tr>
<td>Display type</td>
<td>The type of field in which to display the segment.</td>
</tr>
<tr>
<td>Checked and unchecked values</td>
<td>If the display type is check box, the actual values to save. For example, Y and N or 0 and 1.</td>
</tr>
<tr>
<td>Display size</td>
<td>The character width of the field.</td>
</tr>
<tr>
<td>Display height</td>
<td>The height of the field as measured in visible number of lines when the display type is a text area.</td>
</tr>
<tr>
<td>Read only</td>
<td>Whether the field should display as read-only, not editable text.</td>
</tr>
<tr>
<td>Description help text</td>
<td>The field-level description help text to display for the field. Use description help text to display a field-level description that expands on or clarifies the prompt provided for the field. If description help text is specified, a Help icon button is displayed next to the field in the run time application. The description help text is displayed when the user hovers over the Help icon button.</td>
</tr>
<tr>
<td>Instruction help text</td>
<td>The field-level instruction help text to display for the field. Use instruction help text to provide directions on using the field. If instruction help text is specified, it is displayed in an in-field help note window that appears when users give focus to or hover over the field.</td>
</tr>
</tbody>
</table>
Properties Related to Search

Extensible flexfield segments can be marked as selectively required in search using the indexed property. The indexed property requires end users to enter a value before conducting a search on the attribute represented by the indexed segment. A database administrator must create an index on the segment column representing the indexed attribute.

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. Therefore, 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 of the same data type.
- Both segments must be parts of the same structure in a key flexfield or parts 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 that 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.

Rule Validation of Segment Values

Validation rules on descriptive and extensible flexfield segments determine how an attribute is validated. The value entered for an attribute on a business object may need to match a specified format or be restricted to a list of values. Use a value set to specify the validation rules.

Value set validation is required for global segments and context-sensitive segments, and optional for context segments. In the case of context segments, the application may validate an input value instead of the value set validating the input value against the context segment. However, the application input values must match exactly the valid context segment values. If the context segment values are a superset or subset of the input values, you must assign a table-validated value set or independent value set to validate context values.

When you configure a descriptive flexfield segment, you can specify a constant to use for setting the initial value. The initial value can be an available parameter. For every planned segment, list the constant value or parameter, if any, to use for the initial value.
Naming Conventions

Enter a unique code, name, and description for the segment. These properties are for internal use and not displayed to end users. You can't change the code after the segment is created.

The Application Programming Interface (API) name is a name for the segment that isn't exposed to end users. The API name is used to identify the segment in various integration points including web services, rules, and business intelligence. Use alphanumeric characters only with a leading character. For example, enter a code consisting of the characters A-Z, a-z, 0-9 with a non-numeric leading character. The use of spaces, underscores, multi-byte characters, and leading numeric characters isn't permitted. You can't change the API name after the segment has been created.

Flexfields Segments: How They Are Rendered

Flexfield segments appear on pages as attributes of business objects.

Settings That Affect Flexfield Segment Display

When you configure flexfield segments, the value you enter for the segment's display type determines how the segment appears on the run time page.

How Display Type Values Appear

The figure shows how display types appear at run time.

In the following figure, identify the display type by letter when referring to the table of descriptions for check box, drop-down list, list of values, pop-up list of values, and radio button group.

A. Check Box

B. Drop-down List

C. List of Values

D. Pop-up List of Values

In the following figure, identify the display type by letter when referring to the table of descriptions for radio button group, text area, text box, and date/time.
The table describes each display type. The Example column refers to the figures above.

<table>
<thead>
<tr>
<th>Display Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Box</td>
<td>A</td>
<td>The field is displayed as a check box. If the end user selects the checkbox, the checked value is used. Otherwise, the unchecked value is used.</td>
</tr>
<tr>
<td>Drop-down List</td>
<td>B</td>
<td>The field displays a dropdown list of values from which the end user can select a value.</td>
</tr>
<tr>
<td>List of Values</td>
<td>C</td>
<td>The field displays a dropdown list of values from which the end user can select a value. The user can also click Search to find more values.</td>
</tr>
<tr>
<td>Pop-up List of Values</td>
<td>D</td>
<td>The field displays as a text field with a Search icon button. The end users can type a value in the text field or they can click the Search icon button to open a subwindow for searching.</td>
</tr>
<tr>
<td>Radio Button Group</td>
<td>E</td>
<td>The field is displayed as a set of radio buttons. The end user can select one button. Selecting a button deselects any previously selected button in the set.</td>
</tr>
<tr>
<td>Field Type</td>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>Text Area</td>
<td>F</td>
<td>The field is displayed as a text area in which the end user can type multiple lines of text. The display width and height specify the visible width and number of lines in the text area, respectively.</td>
</tr>
<tr>
<td>Text Box</td>
<td>G</td>
<td>The field is displayed as a text field in which the end user can type a single line of text. The display width controls the width of the text box.</td>
</tr>
<tr>
<td>Date/Time</td>
<td>H</td>
<td>The field enables the end user to enter a date if the data type is Date, or a date and time if the data type is Date Time. The user can select the date from a calendar. If the data type is Date Time, the field also displays fields for specifying the hour, minutes, seconds, AM or PM, and time zone.</td>
</tr>
<tr>
<td>Hidden</td>
<td></td>
<td>The field isn’t displayed.</td>
</tr>
</tbody>
</table>

**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 that 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.

The following aspects are important in understanding how flexfields and value sets work together:

- Defining value sets
- Shared value sets
- Deployment

**Defining Value Sets**

As a key flexfield guideline, define value sets before configuring the flexfield, because you assign value sets to each segment as you configure a flexfield. With descriptive and extensible flexfields, you can define value sets when adding or editing a segment.

**Caution**

Be sure that changes to a shared value set are compatible with all flexfield segments that use 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 value 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.

**Value Set Values**

To configure custom attributes to be captured on the value set values screen in the Manage Value Sets task, configure the Value Set Values descriptive flexfield. The object's code is FND_VS_VALUES_B. This flexfield expects the context code to correspond to the value set code. For each value set, you can define a context whose code is the value set code, and whose context-sensitive segments will be shown for the values of that value set. By default the context segment is hidden since it defaults to the value set code and is not expected to be changed.

You can also define global segments that will be shown for all value sets. However, this would be quite unusual since it would mean that you want to capture that attribute for all values for all value sets.

**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 run time 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 isn't read only, users can modify it. However, if the segment value is a derived value, a user-modified segment value is overwritten when the derivation value changes.

<table>
<thead>
<tr>
<th>Default Type</th>
<th>Default value specified?</th>
<th>Derivation value specified?</th>
<th>Initial run time behavior</th>
<th>Run time behavior after parameter changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>No</td>
<td>Yes</td>
<td>No initial segment value</td>
<td>The changed parameter derivation value updates segment value</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>No</td>
<td>Default segment value</td>
<td>N/A</td>
</tr>
<tr>
<td>----------------</td>
<td>-----</td>
<td>----</td>
<td>------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Constant</td>
<td>Yes</td>
<td>Yes</td>
<td>Default segment value</td>
<td>The changed parameter derivation value updates segment value</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>No</td>
<td>The default segment value is the parameter's default value</td>
<td>N/A</td>
</tr>
<tr>
<td>Parameter</td>
<td>Yes</td>
<td>Yes, and same as default value</td>
<td>The default segment value is the parameter's default and derivation value</td>
<td>The changed parameter derivation value updates segment value</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>Yes, and different from default value</td>
<td>The default segment value is the parameter's default value</td>
<td>The changed parameter default value doesn't update segment value. Only the changed derivation value updates the segment value.</td>
</tr>
</tbody>
</table>

**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 flexfields having segments where the value set is assigned.
FAQs for Flexfield Management

Why did my flexfield changes not appear in the run time 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 sign back in again to view flexfield definition changes reflected in the run time application user interface page.

A flexfield’s deployment status indicates whether the flexfield segments as currently defined in the metadata are available to end users. The flexfield segments seen by end users in the run time correspond to the flexfield definition that was last deployed successfully.

How can I enable flexfield segments for Oracle Social Network Cloud Service?

Descriptive flexfield segments can be enabled for integration with Oracle Social Network Cloud Service. When you manage Oracle Social Network Objects during setup and maintenance, search for the business object that includes descriptive flexfields, and select the business object attributes that are defined as flexfield segments.

Flexfield Deployment

Flexfield Deployment: Explained

Deployment generates or refreshes the Application Development Framework (ADF) business component objects that render the flexfield in a user interface. The deployment process adds the custom attributes to the Web Services Description Language (WSDL) schemas that are exposed by Oracle ADF services and that are used by SOA composites. 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.

If a descriptive flexfield is enabled for business intelligence, the deployment process redeploys the flexfield’s business intelligence artifacts.

You can deploy a flexfield to a sandbox for testing or to the mainline for use in a test or production run time environment. You can deploy extensible flexfields as a background process.

After deployment, the custom attributes are available for incorporating into the SOA infrastructure, such as business process and business rule integration. For example, you can now write business rules that depend on the custom attributes. You must sign out and sign back in to Oracle Fusion Applications to see the changes you deployed in the run time.
The following aspects are important in understanding flexfield deployment:

- Deployment Status
- Initial Deployment Status
- Metadata Validations
- Metadata Synchronization
- Deployment as a Background Process

**Deployment Status**

Every flexfield has a deployment status.

A flexfield can have the following deployment statuses.

<table>
<thead>
<tr>
<th>Deployment Status</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edited</td>
<td>The flexfield metadata definition hasn’t been deployed yet. Updates of the metadata definition aren’t applied in the run time environment yet.</td>
</tr>
<tr>
<td>Patched</td>
<td>The flexfield metadata definition has been modified through a patch or through a data migration action, but the flexfield hasn’t yet been deployed so the updated definition isn’t reflected in the run time environment.</td>
</tr>
<tr>
<td>Deployed to Sandbox</td>
<td>The current metadata for the flexfield is deployed in ADF artifacts and available as a flexfield-enabled sandbox. The status of the sandbox is managed by the Manage Sandboxes task available to the Administrator menu of the Setup and Maintenance work area.</td>
</tr>
<tr>
<td>Deployed</td>
<td>The current metadata for the flexfield is deployed in ADF artifacts and available to end users. There haven’t been any changes to the flexfield since it was last deployed in the mainline.</td>
</tr>
<tr>
<td>Error</td>
<td>The deployment attempt in the mainline failed.</td>
</tr>
</tbody>
</table>

**Note**

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

**Initial Deployment Status of Flexfields**

The Oracle Fusion Applications installation loads flexfield metadata into the database. This initial load sets the flexfield status to Edited. The application provisioning process during installation deploys the 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 run time 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 all flexfield deployment commands. By successfully validating metadata before running the deployment commands, you can avoid failures in the metadata validation phase of a deployment attempt. The deployment process aborts if it encounters an error during the metadata validation phase. Metadata validation results don't affect the deployment status of a flexfield.

**Metadata Synchronization**

When an extensible or descriptive flexfield is deployed, the deployment process regenerates the XML schema definition (XSD), which makes the custom attributes available to web services and the SOA infrastructure.

After deploying a flexfield configuration, you must synchronize the updated XML schema definition (XSD) files in the MDS repositories for each SOA application.

**Note**

To synchronize the updated XSD files in the MDS repositories in Oracle Cloud implementations, log a service request using My Oracle Support at http://support.com/

**Deployment as a Background Process**

You can deploy extensible flexfields or incremental changes made to extensible flexfields as a background process. You must use this action to deploy extensible flexfields that have more than 30 categories. You can also use this action if you want to deploy several extensible flexfields, or if you want to continue working in your session without having to wait for a deployment to complete.

**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.

The following aspects are important in understanding how flexfield deployment status is calculated:

- Settings that affect flexfield deployment status
- How deployment status is calculated
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 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 doesn't affect the deployment status. If the flexfield was deployed to a sandbox and hasn't been edited or redeployed 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 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 that the current flexfield definition isn't 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 aren't yet in effect.

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

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

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

A flexfield-enabled sandbox uses the following components.

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

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

**Sandbox Metadata Services Repository Data**

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

Don’t customize flexfield segment display properties using Page Composer 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. When the flexfield is deployed, the ADF business component objects that implement the flexfield in the run time user interface are generated in the mainline MDS repository from this source.

Deploying a Flexfield to a 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 run time behavior of a flexfield in the flexfield-enabled sandbox. If changes are needed, you 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.

When you deploy a flexfield sandbox, the process generates the name of the flexfield sandbox, and that flexfield sandbox is set as your current active sandbox. When you next sign in to the application, you can see the updated flexfield configurations. The Oracle Fusion Applications global area displays your current session sandbox.

Note

Unlike a standalone sandbox created using the Manage Sandboxes tool, the sandbox deployed for a flexfield contains only the single flexfield. You can manage flexfield sandboxes, such as setting an existing flexfield sandbox as active or deleting it, using the Manage Sandboxes tool.

When you deploy a flexfield to the mainline after having deployed it to the sandbox, the sandbox-enabled flexfield is automatically deleted.

Sandbox MDS Repository Data

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

Warning

Don’t customize flexfield segment display properties using Page Composer 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 and access 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 run time.

You cannot publish the flexfield from the sandbox to the mainline. You must use the Define Flexfields task flow pages to deploy the flexfield for access by users of the mainline because the flexfield configuration in the mainline is the single source of truth.

Deploying Flexfields Using the Command Line: Explained

You can use the Manage Key Flexfields, Manage Descriptive Flexfields, and Manage Extensible Flexfields tasks to deploy flexfields. You can also use WebLogic Server Tool (WLST) commands for priming the Metadata Services (MDS) repository with predefined flexfield artifacts and for deploying flexfields.

The table describes the available commands.

<table>
<thead>
<tr>
<th>WebLogic Server Tool Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deployFlexForApp</td>
<td>Deploys all flexfields for the specified enterprise application. Only flexfields whose status is other than deployed are affected by this command unless the option is enabled to force all flexfields to be deployed regardless of deployment status. Initial application provisioning runs this command to prime the MDS repository with flexfield artifacts.</td>
</tr>
<tr>
<td>deployFlex</td>
<td>Deploy a single flexfield regardless of deployment status</td>
</tr>
<tr>
<td>deployPatchedFlex</td>
<td>Deploys flexfield changes that have been delivered using a flexfield Seed Data Framework (SDF) patch. Deploys flexfields that have a Patched deployment status.</td>
</tr>
<tr>
<td>deleteFlexPatchingLabels</td>
<td>Displays MDS label of flexfield changes for viewing and deleting patching labels.</td>
</tr>
<tr>
<td>validateFlexDeploymentStatus</td>
<td>Displays list containing flexfields that aren’t deployed or failed deployment.</td>
</tr>
</tbody>
</table>
Executing these commands outputs a report at the command line. The report provides the following information for every flexfield that is processed.

- Application identity (APPID)
- Flexfield code
- Deployment result, such as success or error

In case of errors, the report lists the usages for which the errors were encountered. If a run time 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
- Using the `validateFlexDeploymentStatus` 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) if it isn’t 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`. 

16-68 Oracle Product Value Chain Cloud Implementing Product Information Management
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 doesn't deploy flexfields that have a status of Deployed unless the force parameter is set to 'true' (the default setting is 'false').

For more information on priming the MDS partition with configured flexfield artifacts, see the Oracle Fusion Applications Developer's Guide.

From the WLST tool, execute the following commands to deploy the artifacts to the MDS partition, replacing `product_application_shortname` with the application's short name wrapped in single-quotes.

```
deployFlexForApp('product_application_shortname', ['enterprise_id'], ['force'])
```

In a multi-tenant environment, replace `enterprise_id` with the Enterprise ID to which the flexfield is mapped. Otherwise, replace with 'None' or don't 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_shortname', force='true')
```

---

**Tip**

The application's short name is the same as the application's module name.

For more information about working with application taxonomy, see the Oracle Fusion Applications Developer's Guide.

**Using the deployFlex Command**

From the WLST tool, execute the following command to deploy a flexfield, replacing `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. The values must be wrapped in single-quotes.

```
deployFlex('flex_code', 'flex_type')
```
Optionally, execute the following command if the flexfield is an extensible flexfield, and you want to deploy all the flexfield’s configurations.

Note

By default, extensible flexfields are partially deployed. That is, only the pages, contexts, or categories that had recent changes, are deployed.

deployFlex('flex_code', 'flex_type', ['force_Complete_EFF_Deployment'])
where, forceCompleteEFFDeployment=None

Using the deployPatchedFlex Command

Use the deployPatchedFlex command for situations where the patching framework doesn’t 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 isn’t 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.

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 deleteFlexPatchingLabels() 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')

Using the validateFlexDeploymentStatus Command

The validateFlexDeploymentStatus() WLST command checks the deployment status of all flexfields in an Oracle Fusion Applications deployment.

validateFlexDeploymentStatus()

Use this command to verify that all flexfields in the current instance of provisioned Java EE applications are deployed.
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 group of valid values that you assign to a flexfield segment to control the values that are stored for business object attributes.

An end user enters a value for an attribute of a business object while using the application. The flexfield validates the value 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 that changes to a shared value set are compatible with all flexfields segments using the value set.

The following aspects are important in understanding value sets:

- Managing value sets
- Validation
- Security
- Precision and scale
- Usage and deployment

Managing Value Sets

To access the Manage Value Sets page, use the Manage Value Sets task, or use the Manage Descriptive Flexfields and Manage Extensible Flexfields tasks for configuring a segment, including its value set. To access the Manage Values page, select the value set from the Manage Value Sets page, and click Manage Values. Alternatively, click Manage Values from the Edit Value Set page.

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 doesn’t present a list of valid values to users.

**Note**

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

**Security**

Value set security only works in conjunction with usage within flexfield segments.

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. The value set is the resource secured by data security policies. If a value set is secured, every usage of it in any flexfield is secured. It isn't 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 doesn’t 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**

If the data type of a value set is 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 can assign value sets to each segment as you configure a flexfield. With descriptive and extensible flexfield segments, you can create value sets when adding or editing a segment on the runtime page where the flexfield appears.
The following aspects are important in defining value sets:

- Value sets for context segments
- Format-only validation
- Interdependent value sets
- Table validation
- Range
- Security
- Testing and maintenance

**Value Sets for Context Segments**

When assigning a value set to a context segment, you can only use table-validated or independent value sets.

You can use only table and independent value sets to validate context values. The data type must be character and the maximum length of the values being stored must not be larger than the context’s column length. If you use a table value set, the value set cannot reference flexfield segments in the value set’s WHERE clause other than the flexfield segment to which the value set is assigned.

**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, users can enter the value 456 (for a value set with maximum length of three or more), but can’t enter the value ABC. A format only value set doesn’t 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**

Use an independent value set to validate input against a list that isn’t stored in an application table, and not dependent on a subset of another independent value set.

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. 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 don't need to choose a value for another segment first to have access to the subset value set.

Independent, dependent, and subset value sets require a customized list of valid values. Use the Manage Values page to create and manage a value set's valid values and the order in which they appear.

**Tip**

You can customize the Manage Value Sets page to capture additional attributes for each valid value by adding context-sensitive segments in a new context for FND_VS_VALUES_B descriptive field.

## Table Validation

Typically, you use a table-validated set when the values you want to use are already maintained in an application table, such as a table of vendor names. Specify the table column that contains the valid value. You can optionally specify the description and ID columns, a WHERE clause to limit the values to use for your set, and an ORDER BY clause.

If you specify an ID column, then the flexfield saves the ID value, instead of the value from the value column, in the associated flexfield segment. If the underlying table supports translations, you can enable the display of translated text by basing the value set's value column on a translated attribute of the underlying table. You should also define an ID column that is based on an attribute that isn't language-dependent so that the value's invariant ID (an ID that doesn't change) is saved in the transaction table. This allows the run time to display the corresponding translated text from the value column for the run time session's locale.

Table validation lets you enable a segment to depend upon multiple prior segments in the same context structure. You cannot reference other flexfield segments in the table-validated value set's WHERE clause. That is, the WHERE clause cannot reference SEGMENT.segment_code or VALUESET.value_set_code.

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 doesn't 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.

**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.

To enable security on a value set, specify a database resource, typically the code value for the value set. Using the Manage Database Security Policies task, specify conditions, such as filters or SQL predicates, and policies that associate roles with conditions. You can use a filter for simple conditions. For more complex conditions, use a SQL predicate.

Value set data security policies and conditions differ from data security conditions and policies for business objects in the following ways:

- You can grant only read access to end users. You cannot specify any other action.
- When defining a condition that is based on a SQL predicate, use VALUE, VALUE_NUMBER, VALUE_DATE, VALUE_TIMESTAMP, or VALUE_ID to reference the value from a dependent, independent, or subset value set. For table value sets, use a table alias to define the table, such as &TABLE_ALIAS category=70.

When you enable security on 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 shouldn't assume anything about the bind variables; it must assume that 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.

**Testing and Maintenance**

There is no need to define or maintain values for a table-validated value set, as the values are managed as part of the referenced table or independent value set, respectively.

You cannot manage value sets in a sandbox.

When you change an existing value set, the deployment status for all affected flexfields changes to Edited. You must redeploy all flexfields that use that value set to make the flexfields reflect the changes. In the UI pages for managing value sets, the value set's usages show which flexfields are affected by the value set changes.
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 doesn’t require you to provide translated values now, but you cannot change this option if you decide to provide them later.

**Planning Value Sets: Points to Consider**

The value sets you create and configure depend on the valid values on the business object attributes that will use the value set. When creating value sets, you first give the value set a name and description, and then define the valid values of the set.

The following aspects are important in planning value sets:

- List of values
- Plain text input
- Value ranges
- Value format specification
- Security

**List of Values**

You can use one of the following types of lists to specify the valid values for a segment:

- Table column
- Custom list
  - Subset of custom list
- Dependent custom list

If the valid values exist in a table column, use a table value set to specify the list of values. To limit the valid values to a subset of the values in the table, use a SQL WHERE clause. Table value sets also provide some advanced features, such as enabling validation depending on other segments in the same structure.

Use an independent value set to specify a custom set of valid values. For example, you can use an independent value set of Mon, Tue, Wed, and so forth to validate the day of the week. You can also specify a subset of an existing independent value set as the valid values for a segment. For example, if you have an independent value set for the days of the week, then a weekend subset can be composed of entries for Saturday and Sunday.

Use a dependent value set when the available values in the list and the meaning of a given value depend on which independent value was selected for a previously selected segment value. For example, the valid holidays depend on which country you are in. A dependent value set is a collection of value subsets, with one subset for each value in a corresponding independent value set.

For lists of values type value sets, you can additionally limit the valid values that an end user can select or enter by specifying format, minimum value, and
maximum value. For list of values type value sets, you can optionally implement value set data security. If the Oracle Fusion applications are running in different locales, you might need to provide different translations for the values and descriptions.

Plain Text Input

Use a format-only value set when you want to allow end users to enter any value, as long as that value conforms to formatting rules. For example, if you specify a maximum length of 3 and numeric-only, then end users can enter 456, but not 4567 or 45A. You can also specify the minimum and maximum values, whether to right-justify, and whether to zero-fill. With a format-only value set, no other types of validation are applied.

Value Ranges

You can use either a format-only, independent, or dependent value set to specify a range of values. For example, you might create a format-only value set with Number as the format type where the end user can enter only the values between 0 and 100. Or, you might create a format-only value set with Date as the format type where the end user can enter only dates for a specific year, such as a range of 01-JAN-93 to 31-DEC-93. Because the minimum and maximum values enforce these limits, you need not define a value set that contains each of these individual numbers or dates.

Value Format

Flexfield segments commonly require some kind of format specification, regardless of validation type. Before creating a value set, consider how you will specify the required format.

The following table shows options for validation type and value data type.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value data type</td>
<td>Character, Number, Date, Date Time.</td>
</tr>
<tr>
<td>Value subtype</td>
<td>Text, Translated text, Numeric digits only, Time (20:08), Time (20:08:08).</td>
</tr>
<tr>
<td></td>
<td>An additional data type specification for the Character data type for the</td>
</tr>
<tr>
<td></td>
<td>Dependent, Independent, and Format validation types.</td>
</tr>
<tr>
<td>Maximum length</td>
<td>Maximum number of characters or digits for Character data type.</td>
</tr>
<tr>
<td>Precision</td>
<td>Maximum number of digits the user can enter.</td>
</tr>
<tr>
<td>Scale</td>
<td>Maximum number of digits that can follow the decimal point.</td>
</tr>
<tr>
<td>Uppercase only</td>
<td>Lowercase characters automatically changed to uppercase.</td>
</tr>
<tr>
<td>Zero fill</td>
<td>Automatic right-justification and zero-filling of entered numbers (affects</td>
</tr>
<tr>
<td></td>
<td>values that include only the digits 0-9).</td>
</tr>
</tbody>
</table>
Caution

You cannot change the text value data type to a translated text value subtype after creating a value set. If there is any chance you may need to translate displayed values into other languages, choose Translated text. Selecting the Translated text subtype doesn't require you to provide translated values.

Value Sets for Context Segments

You can use only table and independent value sets to validate context values. The data type must be character and the maximum length of the values being stored must not be larger than the context's column length. If you use a table value set, the value set cannot reference flexfield segments in the value set's WHERE clause other than the flexfield segment to which the value set is assigned.

Security

When enabling security on a value set, the data security resource name is an existing value set or one that you want to create. The name typically matches the code value for the value set.

Restriction

You cannot edit the data security resource name after you save your changes.

Table-Validated Value Sets and Bind Variables: Points to Consider

After you assign a value set to a flexfield, you can use bind variables in the WHERE clause.

The following bind variables refer to flexfield elements:

- :{SEGMENT.<segment_code>}
- :{CONTEXT.<context_code>;SEGMENT.<segment_code>}
- :{VALUESET.<value_set_code>}
- :{FLEXFIELD.<internal_code>}
- :{PARAMETER.<parameter_code>}

Segment Code

:SEGMENT.<segment_code>

This bind variable refers to the ID or value of a segment where identifies the segment. Where referring to the ID, the value set is ID-validated. Where referring to the value, the value set isn't ID-validated. The data type of the bind value is the same as the data type of the segment's column.
For both descriptive and extensible flexfields, the segment must be in the same context as the source segment. The source segment contains the WHERE clause. For descriptive flexfields, if the segment is global, then the source segment must be global.

The segment must have a sequence number that is less than the sequence number of the target segment with this bind variable. A matching segment must exist in the current flexfield context.

This bind variable is useful when the set of valid values depends on the value in another segment. For example, the values to select from a CITIES table might depend upon the selected country. If SEGMENT1 contains the country value, then the WHERE clause for the CITIES table might be

\[ \text{<country_code> = :\{SEGMENT.SEGMENT1\}.} \]

**Context Code**

\{:\{CONTEXT.<context_code>;SEGMENT.<segment_code>\}\}

This bind variable, which is valid only for extensible flexfields, refers to the ID (if the value set is ID-validated) or value (if not ID-validated) of a segment that is in a different context than the target segment (the segment with the WHERE clause).

- The `<context_code>` identifies the context and must be in the same category or in an ancestor category. It cannot be a multiple-row context.
- The `<segment_code>` identifies the segment. The data type of the bind value is the same as the data type of the segment's column.

**Tip**

The target segment should appear in the UI after the source segment to ensure the source segment has a value. If the target segment's context is a single-row context, the source and target segments must be on separate pages and the target page must follow the source page.

This bind variable is useful when the set of valid values depends on the value of a segment in another context. For example, the values to select from a CERTIFICATION table for a segment in the Compliance and Certification context might depend on the value of the country segment in the Manufacturing context.

**Value Set Code**

\{:\{VALUESET.<value_set_code>\}\}

This bind variable refers to the ID (if the value set is ID-validated) or value (if not ID-validated) of the segment that is assigned to the value set that is identified by the `<value_set_code>`. The data type of the bind value is the same as the data type of the segment's column.

The segment must have a sequence number that is less than the sequence number of the segment with this bind variable. If more than one segment is assigned to the value set, the closest prior matching segment will be used to resolve the bind expression. A matching segment must exist in the current flexfield context.
This bind variable is useful when the set of valid values depends on the value in another segment and that segment code can vary, such as when the value set is used for more than one context or flexfield. For example, the values to select from a CITIES table might depend upon the selected country. If the value set for the segment that contains the country value is COUNTRIES, then the WHERE clause for the CITIES table might be <county_code> = :{VALUESET.COUNTRIES}.

**Flexfield Internal Code**

:`{FLEXFIELD.<internal_code>}`

This bind variable refers to an internal code of the flexfield in which the value set is used, or to a validation date. The internal_code must be one of the following:

- **APPLICATION_ID** - the application ID of the flexfield in which this value set is used. The data type of APPLICATION_ID and its resulting bind value is NUMBER.
- **DESCRIPTIVE_FLEXFIELD_CODE** - the identifying code of the flexfield in which this value set is used. The data type of DESRIPTIVE_FLEXFIELD_CODE and its resulting bind value is VARCHAR2. Note that you use this string for both descriptive and extensible flexfields.
- **CONTEXT_CODE** - the context code of the flexfield context in which this value set is used. The data type of CONTEXT_CODE and its resulting bind value is VARCHAR2.
- **SEGMENT_CODE** - the identifying code of the flexfield segment in which this value set is used. The data type of SEGMENT_CODE and its resulting bind value is VARCHAR2.
- **VALIDATION_DATE** - the current database date. The data type of VALIDATION_DATE and its resulting bind value is DATE.

**Flexfield Parameters**

:`{PARAMETER.<parameter_code>}`

This bind variable refers to the value of a flexfield parameter where parameter_code identifies the parameter. The data type of the resulting bind value is the same as the parameter’s data type.

---

**Note**

You cannot assign a table value set to a context segment if the WHERE clause uses VALUESET.value_set_code or SEGMENT.segment_code bind variables.

---

**Table-Validated Value Set: Worked Example**

In an application user interface, you want to display a list of values that allow customers to enter satisfaction scores. The value column name is 1, 2, 3, 4, 5 and the value column description is Extremely Satisfied, Satisfied, and so on. Users can pick the appropriate value or description which stores the corresponding name so the name value can be used in a calculation expression.
In this case, you can use the FND LOOKUPS table as the basis for a table-valuated value set. The lookup meaning corresponds to the Value Column Name and the lookup description corresponds to the Description Column Name. The properties of the value set are as follows:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FROM clause</td>
<td>FND_LOOKUPS</td>
</tr>
<tr>
<td>WHERE clause</td>
<td>lookup_type = 'CN_XX_CUST_SATISFACT_SCORE'</td>
</tr>
<tr>
<td>ID column</td>
<td>lookup_code</td>
</tr>
<tr>
<td>Value column</td>
<td>meaning</td>
</tr>
<tr>
<td>Description column</td>
<td>description</td>
</tr>
<tr>
<td>Enable Flag column</td>
<td>enabled_flag</td>
</tr>
<tr>
<td>Start Date column</td>
<td>start_date_active</td>
</tr>
<tr>
<td>End Date column</td>
<td>end_date_active</td>
</tr>
<tr>
<td>Order by</td>
<td>display_sequence</td>
</tr>
</tbody>
</table>

After completing this task, you should have created your customer satisfaction value set for the Incentive Compensation page of your implementation project.

**Creating a Value Set Based on a Lookup**

1. From the Setup and Maintenance work area, find the Manage Value Sets task and click the Go to Task icon button.
2. On the Manage Value Sets page, click the Create icon button.
3. On the Create Value Set page, enter the following values:
   a. In the Value Set Code field, enter `CN_XX_CUSTOMER_SATISFACTION_SCORES`
   b. In the Description field, enter Customer satisfaction score.
   c. In the Module field, select Search....
   d. In the Search and Select: Module subwindow, enter Incent in the User Module Name field
   e. Select Incentive Compensation.
   f. Click OK.
4. On the Create Value Set page, enter the following values:
   a. In the Validation Type field, select Table.
   b. In the Value Data Type field, select Character.
   c. In the Definition section FROM Clause field, enter FND_LOOKUPS.
   d. In the Value Column Name field, enter DESCRIPTION.
   e. In the Description Column Name field, enter MEANING.
   f. In the ID Column Name field, enter LOOKUP_CODE.
   g. In the Enabled Flag Column Name field, enter 'Y'.
   h. In the Start Date Column Name field, enter START_DATE_ACTIVE.
   i. In the End Date Column Name field, enter END_DATE_ACTIVE.
j. In the WHERE Clause field, enter LOOKUP_TYPE = 'CN_xx_CUST_SATISFACT_SCORE'.

5. Click Save and Close.

6. In the Manage Value Sets page, click Done.

**Adding Attributes to the Manage Value Sets Page: Procedures**

For independent, dependent, and subset value sets, you can add attributes to a value set. The attributes appear in the Manage Value Sets UI for capturing additional information about each valid value, such as its purpose.

Typically, these attributes are used to capture internal information. To display attributes on an application page, you must programmatically modify the application to access them.

1. Find the FND_VS_VALUES_B flexfield using the Manage Descriptive Flexfields task.
2. Open FND_VS_VALUES_B for editing.
3. Click Manage Contexts.
4. Create a new context and use the value set code for the context code.
5. Add the new attributes as context-sensitive segments.
6. Deploy FND_VS_VALUES_B to the run time.
7. Sign out and sign back in.
8. Open the Manage Value Sets page to view the new attributes.

**FAQs for Manage Value Sets**

**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 for a descriptive flexfield segment to be a parameter, which means the entity object attribute to which the chosen parameter is mapped provides the initial default value for the segment.

You can set the default value 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 run time 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.

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 aren’t 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 whether or not to configure and expose segments for the descriptive flexfield defined and registered in your database. For lists of descriptive flexfields, see assets with the Flexfield: Descriptive type in Oracle Enterprise Repository for Oracle Fusion Applications (http://fusionappsoer.oracle.com).

A descriptive flexfield provides a set amount of segments for an entity. You make the segments of a descriptive flexfield 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.
<table>
<thead>
<tr>
<th>Segment Type</th>
<th>Run Time Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global segment</td>
<td>Always available</td>
</tr>
<tr>
<td>Context segment</td>
<td>Determines which context-sensitive segments are displayed</td>
</tr>
<tr>
<td>Context-sensitive segment</td>
<td>Displayed depending on the value of the context segment</td>
</tr>
</tbody>
</table>

In the figure, a descriptive flexfield has one context segment called Category for which there are three values: Resistor, Battery, and Capacitor. In addition, the descriptive flexfield consists of two global segments that appear in each of the contexts, and three context-sensitive segments that only appear in the context in which they are configured.

Application development determines the number of segments available for configuring. During implementation, you configure the flexfield by determining the following:

- Which attributes to add using the available segments
- The context values
- 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.
Planning Descriptive Flexfields: Points to Consider

Once you have identified a flexfield to configure, plan the configuration in advance. Compile a list of the UI pages and other artifacts in your deployment that are affected by the configuration. Verify that you are provisioned with the roles needed to view and configure the flexfield. View the flexfield using the Highlight Flexfields command in the Administration menu while viewing the run time page where the flexfield appears. Plan how you will deploy the flexfield for test and production users. Review the tools and tasks available for managing flexfields, such as the Define Flexfields task list, Manage Sandboxes, and Highlight Flexfields for adding and editing flexfield segments.

Planning a descriptive flexfield can involve the following tasks:

1. Identify existing parameters.
2. Identify existing context values and whether the context value is derived.
3. Identify custom attributes and plan the descriptive flexfield segments, segment properties, and structure.
5. Plan initial values.
6. Plan attribute mapping to Oracle Business Intelligence objects.

Identify Existing Descriptive Flexfield Parameters

Some descriptive flexfields provide parameters that can be used to specify the initial value of a descriptive flexfield segment. The parameter is external reference data, such as a column value or a session variable. For example, if a flexfield has a user email parameter, you can configure the initial value for a customer email attribute to be derived from that parameter.

Review the list of available parameters in the Derivation Value field in the Create Segment page for a descriptive flexfield. If you decide to use one of the parameters to set an initial value, select that parameter from the Derivation Value drop-down list when you add the descriptive flexfield segment.

Evaluate Whether the Context Value Is Derived

The context value for a descriptive flexfield might have been preconfigured to be derived from an external reference. For example, if the context is Marriage Status, then the value might be derived from an attribute in the employee business object. When the context value is derived, you might need to take the derived values and their source into consideration in your plan.

To determine whether the context value is derived, access the Edit Descriptive Flexfield task to view the list of configured context values for the flexfield. The Derivation Value field in the Context Segment region displays a list of available parameters.

If context values have been preconfigured, see Oracle Fusion Applications Help for product-specific information about the use of those values.
Plan the Segments, Segment Properties, and Structure

Identify the custom attributes you need for a business object to determine the segments of the descriptive flexfield. Determine the segment properties such as the prompt, display type, or initial value.

The structure of the descriptive flexfield is determined by its global, context, and context-sensitive segments. Plan a global segment that captures an attribute for every instance of the business object. Plan a context for segments that depend on a condition of situation applying to a particular instance of the business object. Plan context-sensitive segments to capture attributes that are relevant in the context.

There is only one context segment available for descriptive flexfields. If you have more than one group of custom attributes where you could use the context segment, you will have to pick one group over the others, based on your company’s needs and priorities, and add the other custom attributes as global segments.

Plan Validation Rules

Define each segment’s validation rules and check if value sets exist for those rules or you must create new ones. If you must create a value set, you can create it either before configuring the flexfield or while creating or editing a segment.

When determining a segment’s validation rules, consider the following questions:

- What is the data type - character, date, date and time, or number?

- Does the segment require any validation beyond data type and maximum length?

- Should a character type value be restricted to digits, or are alphabetic characters allowed?

- Should alphabetic characters automatically be changed to uppercase?

- Should numeric values be zero-filled?

- How many digits can follow the radix separator of a numeric value? In base ten numerical systems the radix separator is decimal point.

- Does the value need to fall within a range?

- Should the value be selected from a list of valid values? If so, consider the following questions:
  - Can you use an existing application table from which to obtain the list of valid values, or do you need to create a custom list?
  - If you are using an existing table, do you need to limit the list of values using a WHERE clause?
  - Does the list of valid values depend on the value in another flexfield segment?
• Is the list of valid values a subset of another flexfield segment's list of values?

**Plan Initial Values**

For every segment, list the constant value or SQL statement, if any, to use for the initial value of the custom attribute.

**Plan How Segments Map to Oracle Business Intelligence Objects**

If a descriptive flexfield has been enabled for Oracle Business Intelligence, you can make it available for use in Oracle Business Intelligence applications. You can use segment labels to map segments to logical objects. Plan to map segments to logical objects before deploying the flexfield as a way to streamline the import into Oracle Business Intelligence.

Use the Manage Segment Labels page to view preconfigured segment labels. If a segment label doesn’t exist for the logical object, then decide on a code, name, and description in preparation for adding that label. Choose a code, name, and description that will help end users select the correct label.

The mapping equalizes similar context-sensitive attributes that are from different contexts but are mapped to a single logical object. For information about objects in the logical model, see the “Working with Logical Tables, Joins, and Columns” chapter in the Oracle Fusion Middleware Metadata Repository Builder’s Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition).

**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.

In general, configuring descriptive flexfields involves:

1. Creating segment labels for business intelligence enabled flexfields.
2. Configuring global segments by providing identity information, the initial default value, and the display properties.
3. Configuring the context segment by specifying the prompt, whether the context segment should be displayed, and whether a value is required.
4. Configuring contexts by specifying a context code, description, and name for each context value, and adding its context-sensitive segments, each of which is configured to include identifying information, the column assignment, the initial default value, and the display properties.

The following aspects are important in understanding descriptive flexfield management:
• Segments
• Adding Segments to a Highlighted Flexfield
• Usages
• Parameters
• Delimiters
• Initial Values
• Business Intelligence

**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. The value set must be independent or table-validated. If table-validated, the WHERE clause must not use the VALUESET.value_set_code or SEGMENT.segment_code bind variables. The value set must be of data type Character with the maximum length of values being stored no larger than the context's column length.

If you don't 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.

**Adding Segments to a Highlighted Flexfield**

When you highlight flexfields on a run time page and use an Add Segment icon button to create a segment, the segment code, name, description, table column, and sequence number are set automatically. If you use an Add Segment icon button to configure descriptive flexfield segments, you cannot use an existing value set. Value sets are created automatically when you add the segments. You can enter the valid values, their descriptions, and the default value or specify the formatting constraints for the value set, such as minimum and maximum values.

Depending on display type, the value set you create with the Add Segment icon button is either an independent value set or a format-only value set. The
table shows which type of value set is created depending on the segment display component you select.

<table>
<thead>
<tr>
<th>Display Component</th>
<th>Value Set Created with Add Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check box</td>
<td>Independent</td>
</tr>
<tr>
<td>Drop-down list</td>
<td>Independent</td>
</tr>
<tr>
<td>List of Values</td>
<td>Independent</td>
</tr>
<tr>
<td>Radio Button Group</td>
<td>Independent</td>
</tr>
<tr>
<td>Text Field With Search</td>
<td>Independent</td>
</tr>
<tr>
<td>Text box</td>
<td>Format Only</td>
</tr>
<tr>
<td>Text area</td>
<td>Format Only</td>
</tr>
<tr>
<td>Date/Time</td>
<td>Format Only</td>
</tr>
</tbody>
</table>

**Tip**

After you add a context value, refresh the page to see the new value.

**Usages**

Descriptive flexfield usages allow for the same definition to be applied to multiple entities or application tables, such as a USER table and a USER_HISTORY table. Descriptive flexfield tables define the placeholder entity where the flexfield segment values are stored once you have configured the descriptive flexfield. When you configure a flexfield, the configuration applies to all its usages.

**Parameters**

Some descriptive flexfields provide parameters, which are attributes of the same or related entity objects. Parameters are public arguments to a descriptive flexfield. Parameters provide outside values in descriptive flexfield validation. 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. Parameters can be referenced by the logic that derives the default segment value, and by 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.

**Initial Values**

The SQL statement defining an initial value must be a valid statement that returns only one row and a value of the correct type.

You can use two types of SQL statements:
- SQL statement with no binding. For example, select MIN(SALARY) from EMPLOYEES.

- SQL statement with bind variables. You can use the following bind variables in the WHERE clause of the SQL statement.

  - \{(\text{SEGMENT.<segment_code>})\}: Identifies a segment in the same context.
  - \{(\text{CONTEXT.<context_code>};\text{SEGMENT.<segment_code>})\}: Identifies a segment in a different context. The context must be in the same category or in an ancestor category, and it cannot be a multiple-row context.
  - \{(\text{VALUESET.<value_set_code>})\}: Identifies the closest prior segment in the same context that is assigned to the specified value set.
  - \{(\text{FLEXFIELD.<internal_code>})\}: Identifies a flexfield.

For more information about using bind variables, see the help for value sets.

**Business Intelligence**

Selecting a global, context, or context-sensitive segment’s BI Enabled checkbox specifies that the segment is available for use in Oracle Business Intelligence.

When the flexfield is imported into Oracle Business Intelligence, the label you selected from the BI Label dropdown list equalizes the segment with segments in other contexts, and maps the segment to the logical object represented by the label.

**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 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, use the Import Oracle Fusion Data Extensions for Transactional Business Intelligence process to import the flexfield changes into the Oracle Business Intelligence repository. Users can make use of the newly-generated attributes in business intelligence applications.
For example, a user can generate a report that includes attributes added by the descriptive flexfield. For additional information about logical objects and import, refer to the Oracle Transactional Business Intelligence Administrator’s Guide.

**Flattening**

When you deploy a business intelligence-enabled descriptive flexfield, the deployment process generates an additional set of flattened Application Development Framework (ADF) business components in addition to the usual ADF business components and ADF faces run time artifacts that are generated during deployment. The flattened business components include attributes for business intelligence-enabled segments only. Flattening means each custom column in each context shows up as an attribute in an Oracle Business Intelligence folder.

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, consider the following:

- If you assigned a label to the segment, the flattened components include an additional single attribute representing segments with that label.
- If you didn’t assign a label, the flattened components include a discrete attribute for each BI-enabled context-sensitive segment in each context.

**Mapping to Logical Objects in Business Intelligence**

You can simplify reporting by representing similar segments as a single logical object in Business Intelligence.

If you assign a label to any set of context-sensitive segments that serve the same purpose in different contexts, you can consolidate or equalize the segments into a single attribute. This prevents duplication and the extra workload and complexity that result from the flattening process. For example, a United States context might have a Passport segment and a Canada context might have Visa segment. If you assign the NationalID segment label to both the Passport and Visa segments, they are equalized into the same NationalID attribute in the flattened business component.

Non-labeled context-sensitive segments aren’t 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.

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 doesn’t 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.

**Importing to Oracle Business Intelligence Repository**

After you deploy a business intelligence-enabled flexfield, import the flexfield changes into the Oracle Business Intelligence repository to make use of the newly flattened business components in business intelligence and then propagate the flexfield object changes. When you import the metadata into the Oracle Business Intelligence repository, you must do so as the FUSION_APPS_BI_APPID user.

---

**Note**

To import flexfield changes into the Oracle Business Intelligence repository in Oracle Cloud implementations, run the Import Oracle Fusion Data Extensions for Transactional Business Intelligence process. For additional information about import, refer to the Oracle Transactional Business Intelligence Administrator's Guide.

---

**Tip**

When you import a flexfield into the Oracle Business Intelligence repository, you see both `<name>` and `<name>_c` attributes for each segment, along with some other optional attributes. The `<name>` attribute contains the value. The `<name>_c` attribute contains the code of the value set that the value comes from, and is used for linking to the value dimension. You must import both attributes.

---

**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 aren’t restricted by the number of columns 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.

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 doesn’t differ by context value.

An extensible flexfield describes an application entity, with the run time 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.

For lists of extensible flexfields, see assets with the Flexfield: Extensible type in Oracle Enterprise Repository for Oracle Fusion Applications (http://fusionappsoer.oracle.com).

The following aspects are important in understanding key flexfields:

• Usages
• Categories
• Pages
• Security

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.

Usages include security information for applying no security to user access or enforcing view and edit privileges. Some product-specific extensible flexfields have special usage fields beyond those for security.

**Categories**

You can configure multiple extensible flexfield contexts and group the contexts into categories. All extensible flexfields have at least one category. For some extensible flexfields, you can configure a hierarchy of categories. A child category in the hierarchy can inherit contexts from its parent category.

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 let you 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.

**Security**

When you configure a flexfield, you set the privileges for a context at the usage level by selecting actions for the view and edit privileges of a context usage.

When an end user performs a search, the user interface displays only the attribute values of the contexts for which the user has view privileges. The user is able to perform a search using all attributes for all contexts, regardless of view privileges.

If end users access a context through a web service, an exception is thrown if they perform an action for which they don't have privileges.

All extensible flexfields have a base data security resource. Some data security resources for extensible flexfields are preconfigured with actions that you can use to specify access privileges. If no action is preconfigured, a security administrator can create actions and policies to support access control on the extensible flexfield attributes.
Some extensible flexfields have a translatable option; these flexfields also have a translation data security resource.

**Managing Extensible Flexfields: Points to Consider**

Configuring extensible flexfields involves managing the available flexfields registered with your application database.

The following sequence describes how to configure extensible flexfields:

1. Configuring contexts by creating each context segment and the context-sensitive segments for each context segment, and providing the following for each segments:
   a. Identifying information
   b. Column assignment
   c. Initial default value
   d. Display properties
2. Configuring context usages and usage security by selecting actions to which users should have access:
   - View
   - Edit
   - None, if no special privileges should be enforced.
3. Configuring categories and category details.
4. Associating contexts with a category.
5. Creating logical pages for a category.

The following aspects are important in understanding extensible flexfield management:

- Contexts
- Categories
- Initial values
- Indexed segments
- Pages
- Security
- Deployment

**Contexts**

Each context is displayed to end users as a region containing its context-sensitive segments. You can specify instruction help text to display instructions that explain how to use the region and its attributes to end users. Instruction help text
is displayed at the top of the context region. 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 segment for chapter and a segment for number of pages. Multiple chapters can then be associated with each book in the BOOK table.

For contexts that store multiple rows, you can uniquely identify each row by having the values in each row form a unique key.

If flexfield has a category hierarchy, then you can leverage the hierarchy to reuse contexts for similar entities, such as similar items in a product catalog.

Set the context to translatable so that 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.

Set the context security to give an end user view or edit access to a context. The context’s task flow and region appear in the user interface only for users with view access. With edit access, an end user can edit the context’s attribute values. With no action specified for a usage, no special privileges are enforced through the context’s configuration.

**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. Extensible flexfields with more than 30 categories must be deployed as a background process.

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. Consider category hierarchies to be 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 various 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 logical pages, and determine the sequence in which the pages appear. This serves to connect the contexts so they will always be presented together and in a particular order 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.

**Initial Values**

The SQL statement defining an initial value must be a valid statement that returns only one row and a value of the correct type.

You can use two types of SQL statements:

- SQL statement with no binding. For example, select MIN(SALARY) from EMPLOYEES.
- SQL statement with bind variables. You can use the following bind variables in the WHERE clause of the SQL statement.
  - `:{SEGMENT.<segment_code>}`: Identifies a segment in the same context.
  - `:{CONTEXT.<context_code>};SEGMENT.<segment_code>`: Identifies a segment in a different context. The context must be in the same category or in an ancestor category, and it cannot be a multiple-row context.
  - `:{VALUESET.<value_set_code>}`: Identifies the closest prior segment in the same context that is assigned to the specified value set.
  - `:{FLEXFIELD.<internal_code>}`: Identifies a flexfield.

For more information about using bind variables, see the help for value sets.

**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. Commonly, a database administrator (DBA) adds columns to the database index.

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 criteria. No search is performed if an end user enters an attribute that isn't indexed as a search criterion.

**Pages**

Define logical pages to group contexts together in the user interface. For a given category, you may create one or more logical pages. You may add one or more of the category’s associated contexts to each of the category’s logical pages.

You can specify:

- The sequence of the contexts within each page.
- The sequence in which the logical pages appear.
- Instruction help text to display instructions that explain how to use the page to end users. Instruction help text is displayed at the top of the logical page, preceding all of its context regions.

**Security**

An extensible flexfield’s base data security resource typically has a name with an _B suffix. The translation data security resource is a view of a translation table that typically has a name with an _VL suffix.

If a flexfield supports the translatable option and has a translation data security resource, make sure that you create the action for the appropriate data security resource.

- If you create a context-specific action for a nontranslatable context, add it to the base data security resource.
- If you create a context-specific action for a translatable context, add it to the translation data security resource.

**Deployment**

You can only deploy extensible flexfields using the Manage Extensible Flexfields task. You can deploy extensible flexfields offline as a background process and continue working in the session without having to wait for the deployment to complete. You can add one after another extensible flexfield to your deployment queue when you deploy as a background process. The flexfields are deployed, one at a time, in the order that you deploy them to the queue. You must deploy extensible flexfields with more than 30 categories as a background process.

You can remove an extensible flexfield from the deployment queue with the Cancel Background Deployment command.
When an extensible flexfield is deployed in a background process, its offline status indicates that the flexfield is in a background deployment process. A flexfield’s offline status is cleared and its deployment status updated when the background deployment process has completed.

**Note**

The **Offline Status** column refreshes when you perform a new search in the Manage Extensible Flexfields task.

---

**Configuring an Item Extended Attributes Flexfield: Example**

The Item Extended Attributes flexfield provides segments for extending the Item business object. In the Manage Extensible Flexfields task, you configure your product business object to include a Technical Specifications logical page in the user interface for the Electronics and Computers category of items.

In this example, your configuration of this flexfield groups custom attributes into the following contexts:

- Materials and Substances
- Compliance and Certification
- Voltage

**Scenario**

The following list shows an example plan for custom computer attributes for the Item Extended Attributes flexfield. In this example, the Electronics Information page is inherited from the parent Electronics and Computers category.

- Page: Electronics Information
  - Context: Compliance and Certification, single row
    - ISO 14001 (International Organization for Standardization for an Environmental Management System)
    - ENERGY STAR (energy efficiency guidelines)
    - ROHS (Restriction of the use of certain hazardous substances in electrical and electronic equipment)
  - Context: Voltage, single row
    - Minimum voltage
    - Maximum voltage
    - Current type
  - Context: Materials and Substances, multiple rows
    - Material
    - Contain recyclate
• Percent unit mass
• Page: Computer Information
• Context: Processor Specifications, single row
• Manufacturer
• CPU type
• Processor interface
• Processor class
• Processor speed
• Cores

The following table summarizes key decisions for this scenario.

<table>
<thead>
<tr>
<th>Decisions to Consider</th>
<th>In This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Which extensible flexfield is available for configuring a hierarchy of categories?</td>
<td>Item Extended Attributes flexfield</td>
</tr>
</tbody>
</table>

**Collecting Technical Specifications**

Your product inventory pages for electronics and computers require a technical specifications page. Your product inventory pages for furniture require a furniture specifications page and an assembly instructions page. Items in both the electronics and computer category, and in the furniture category, share attributes for specifying materials and substances.

The figure shows a Technical Specifications logical page in the user interface for the Electronics and Computers category, with attributes in the context of Recovery and Recycling, Compliance and Certification, Operating Conditions, and Materials and Substances. The Materials and Substances context is configured for multiple rows so your users can select all the materials and substances required to make a single product, displayed as attribute values in a table.
Analysis

You use logical pages to arrange how the contexts appear in the user interface. Use a context to store all the materials and substances required to make a single product. You can configure a context to store multiple rows per entity. The multiple rows are displayed in a table, as for the Materials and Substances context.

The Technical Specifications logical page contains the attributes for the four contexts.

- Recovery and Recycling
- Compliance and Certification
- Operating Conditions
- Materials and Substances

In the figure, the Furniture category is configured to include a Furniture Specifications logical page and an Assembly Instructions logical page. The two categories (Electronics & Computers and Furniture) share the Materials & Substances context.

Configure Security for the Item Flexfield Configuration

The following table shows an example of data security policies for the Item flexfield.

<table>
<thead>
<tr>
<th>Data Security Resource</th>
<th>Policy</th>
<th>Role</th>
<th>Actions</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITEM_EFF_B</td>
<td>A</td>
<td>VOLTAGE_SPEC</td>
<td>edit_nontrans_voltag</td>
<td>All values</td>
</tr>
<tr>
<td>ITEM_EFF_VL</td>
<td>B</td>
<td>COMPLIANCE_SPEC</td>
<td>edit_trans_compliance</td>
<td>All values</td>
</tr>
<tr>
<td>ITEM_EFF_VL</td>
<td>C</td>
<td>COMPUTER_SPEC</td>
<td>edit_trans_attrs</td>
<td>ComputerCategoryFilter</td>
</tr>
<tr>
<td>ITEM_EFF_VL</td>
<td>D</td>
<td>TELEVISION_SPEC</td>
<td>edit_trans_attrs</td>
<td>TVCategoryFilter</td>
</tr>
</tbody>
</table>

The following table shows the privileges for three of the flexfield’s contexts.
In this example, anyone can view the contexts’ attributes, but the edit privileges are restricted as follows:

- Voltage: Editable only by voltage specialists.
- Compliance and Certification: Editable only by compliance specialists.
- Materials and Substances: Only computer specialists can edit these attributes for items in the computer category. Only television specialists can edit these attributes for items in the TV category.

In this example, the Materials and Substances context is secured by a generic action with a condition applied to restrict access by category. Voltage and Compliance and Certification are secured by actions specific to each context.

**FAQs for Manage Extensible Flexfields**

**Why did the extensible flexfield context not appear at run time?**

If a deployed extensible flexfield context doesn’t appear in the user interface, verify that the context is associated with one of the category’s pages defined for the extensible flexfield.

**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 sees 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 aren’t 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.

For lists of key flexfields, see assets with the Flexfield: Key type in Oracle Enterprise Repository for Oracle Fusion Applications (http://fusionappsoer.oracle.com).

**Note**

For information about specific key flexfields, see the Oracle Fusion Applications Help for the product where the associated business component is implemented.

**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. Flexfield metadata is stored in flexfield metadata tables.

Based on the flexfield metadata, actual part numbers are captured at run time 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 combinations 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
Define Applications Core Configuration

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 have ten parts, define ten combinations. A valid combination is simply an existing or new combination that can currently be used because it isn’t out of date or disabled, and doesn’t 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 lets you 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. Maintain individual combinations, such as part numbers in the combinations page.

**Dynamic Combination Creation**

Dynamic combination creation is the insertion of a new valid combination into a combinations table from a page other than the combinations page.

Dynamic combination creation may be enabled at the following levels.

<table>
<thead>
<tr>
<th>Level Of Dynamic Combination Creation</th>
<th>Controlled By:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexfield</td>
<td>Application development</td>
</tr>
</tbody>
</table>
If your key flexfield or certain usages or references of the key flexfield don’t 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 doesn’t 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 isn’t the underlying table for the foreign key page.

Planning Key Flexfields: Points to Consider

Your first step in planning your key flexfields is to determine which key flexfields your application requires.

Your plan should include:

- The purpose of the key flexfield
- The number and length of its available segment columns
- Whether your key flexfield allows more than one structure
- Whether more than one structure must be defined
- The number, order and length of your segments for each structure

Consider the following aspects in planning flexfields:

- Before you begin
- Access to flexfield-related tasks
- Restrictions
- Validation rules for flexfield segments

Before You Begin

Once you have identified a flexfield to configure, plan the configuration in advance. Compile a list of the UI pages and other artifacts in your deployment that are affected by the configuration. Verify that you are provisioned with the roles needed to view and configure the flexfield. View the flexfield using the Highlight Flexfields command in the Administration menu while viewing the run time page where the flexfield appears. Plan how you will deploy the flexfield for test and production users.
Review the tools and tasks available for managing flexfields, such as the Define Flexfields task list and Manage Sandboxes.

If you plan to use value sets, create them before configuring the key flexfield. You cannot create value sets for key flexfields at the time that you add and configure key flexfield segments.

**Access to Flexfield-Related Tasks**

To access tasks for configuring flexfields and value sets, you must be provisioned with roles that entitle you to access the tasks in the Define Flexfields task list or tasks for managing product-specific flexfields. Contact your security administrator for details. For information about product-specific flexfield tasks, such as Manage Fixed Assets Key Flexfields, consult the product-specific documentation in Oracle Fusion Applications Help.

**Restrictions**

If you plan to use value sets, create them before configuring the flexfield.

Plan your key flexfield configuration to scale to your enterprise needs. For example, if you expect to disable old cost centers and enable new ones frequently, plan a larger maximum size for your cost center value set so that you can have more available values. One thousand available values for a 3-character value set provides more room for changes than 100 available values for a 2-character value set.

Note the code name of the flexfield you intend to configure so you can find it easily in the Define Flexfield task list or tasks for managing product-specific key flexfields.

In some cases you can customize how the flexfield appears on the page.

See Oracle Fusion Applications Help for specific products to determine any restrictions on using product-specific key flexfields.

**Reporting**

If you want to report on your data by certain criteria or sub-entities, such as account number or project or region, consider making that sub-entity a distinct segment, rather than combining it with another sub-entity, so that you can categorize and report on smaller discrete units of information.

**Managing Key Flexfields: Points to Consider**

Consider the plans for a key flexfield, security, and resulting run time pages when configuring key flexfields.

**Planning**

Plan structures carefully and allow for future needs.

---

16-108 Oracle Product Value Chain Cloud Implementing Product Information Management
Don’t 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 doesn’t conflict with the flexfield data. For example, if your data frequently contains periods, such as in monetary or numeric values, don’t use a period as your segment separator. Any character you expect to appear frequently in your segment values or descriptions isn’t 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 combinations 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.

**Run time 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 the 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 pages 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 lets you manipulate 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 enables the key flexfield to behave more like a descriptive flexfield.

A code combination maintenance page or combinations page presents the combinations table. This enables directly creating and maintaining code combinations. The combinations table contains all key flexfield segment columns and a unique ID column.

A typical application has only one combinations page. An application might not have a combinations page if it doesn’t 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 that you can reuse a single key flexfield in multiple combinations of the same segments or a subset of those 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 doesn’t display disabled segments in run time.

**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

At the structure level, differences among structure instances include whether dynamic combination creation is allowed.

At the structure instance level, differences among segment instances 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 can 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. Commonly, a database administrator (DBA) adds columns to the database index.

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 search criteria. No search is performed if an end user doesn’t enter at least one query required attribute as search criteria.

Tip

Index the Structure Instance Number column on your combinations table to improve run time 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 lets end users enter values at run time that produce new code combinations for the flexfield. If **Dynamic Combination Creation Allowed** isn’t 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 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.

**Cross Validation Rules: Explained**

You can control the creation of new key flexfield code combinations by defining cross validation rules. A cross-validation rule defines validation across segments and enforces whether a value of a particular segment can be combined with specific values of other segments to form a new combination.

The table compares segment validation to cross-segment validation:

<table>
<thead>
<tr>
<th>Type of validation</th>
<th>Type of control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment validation</td>
<td>Controls the values you can enter for a particular segment</td>
</tr>
<tr>
<td>Cross-segment validation</td>
<td>Controls the combinations of values that administrators and end users can create for key flexfields</td>
</tr>
</tbody>
</table>

**Note**

You can use cross-validation rules for any key flexfield that has cross-validation enabled. See the documentation for your key flexfield to determine if it supports cross validation.

Cross-validation rules prevent the creation of combinations with values that shouldn’t coexist in the same combination. For example, your company requires that all revenue accounts must have a specific department. Therefore, account...
combinations that have revenue account values, such as all values between 4000 and 5999, must have a corresponding department value other than 000, which indicates no department is specified. You can define cross validation rules that disallow creation of combinations with incompatible segments, such as 4100-000 or 5000-000.

Alternatively, suppose your accounting key flexfield has an Organization segment with two possible values, 01 and 02. You also have a Natural Account segment with many possible values, but company policy requires that Organization 01 uses the natural account values 001 to 499 and Organization 02 uses the natural account values 500 to 999. You can create cross-validation rules to ensure that users cannot create a general ledger account with combinations of values such as 02-342 or 01-750.

The following aspects are important to understanding cross validation rules:

- Rule Definitions
- Enforcement
- Timing

**Rule Definitions**

Cross validation rules consist of the following information:

<table>
<thead>
<tr>
<th>Rule Feature</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Uniquely identifies cross validation rules in a deployment.</td>
</tr>
<tr>
<td>Description</td>
<td>Helps administrators identify the purpose of the rule.</td>
</tr>
<tr>
<td>Error message</td>
<td>Explains why the attempted combination violates the rule.</td>
</tr>
<tr>
<td>Start Date, End Date</td>
<td>Indicates the period of time when the rule is in effect.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Determines whether the rule is enforced.</td>
</tr>
<tr>
<td>Condition filter</td>
<td>Determines the conditions under which an enabled cross validation rule should be evaluated.</td>
</tr>
<tr>
<td>Validation filter</td>
<td>Determines the validation that the rule enforces when that condition is met.</td>
</tr>
</tbody>
</table>

When the event specified in the condition filter is applicable, the validation filter condition must be satisfied before the combination can be created. If the event specified in the condition filter isn’t applicable, then the combination is considered to pass the rule and the rule won’t be evaluated even if it is enabled.

**Note**

If you don’t specify any statement in the condition filter, then the condition is always true and the rule is always evaluated.

**Enforcement**

Cross-validation prevents creation of invalid combinations by administrators using maintenance pages and end users using dynamic insertion in foreign key pages.
Enabled rules are enforced when there is an attempt to create a new combination of segment values. Disabled rules are ignored. Deleting the rule has the same effect, but you can re-enable a disabled rule.

**Timing**

When users attempt to create a new combination, the key flexfield evaluates any cross-validation rules that are enabled and in effect.

**Warning**

Cross-validation rules have no effect on combinations that already exist. The flexfield treats any existing invalid combinations that pre-date the rule as valid.

If you want to prevent users from using previously existing combinations that are no longer valid according to your cross-validation rules, manually disable those combinations using the combinations page for that key flexfield.

When defining a cross-validation rule, specify a start and end date to limit the time when the rule is in effect. The rule is valid for the time including the From and To dates.

**Cross Validation Rules: Points to Consider**

When you need key flexfield combinations of segment values validated across segments, you can optimize your cross validation rules to improve the experience of administrators and end users.

Consider the following when defining cross validation rules:

- Filters
- Rule Complexity
- Maintenance

**Filters**

A cross validation rule includes a condition filter and a validation filter.

The rule is evaluated using the following logic: If the condition filter is satisfied, then validate that the validation filter is satisfied.

1. The condition filter describes the event under which the rule will be evaluated. If the event specified in the condition filter isn’t applicable, then the rule won’t be evaluated even if it is enabled.

2. When the event specified in the condition filter is applicable, the validation filter condition must be satisfied before the combination can be created.

For example, if your organization has determined that a certain company value, Operations, cannot use a specific cost center, Marketing, you can define a cross validation rule to validate your combinations.

1. The rule evaluates the company condition filter.

2. When company is equal to Operations, the rule evaluates the cost center validation filter.
3. When cost center is equal to Marketing, the rule prevents a combination from being created.

4. The error message you defined for the rule displays to inform the user that the attempted combination violates the rule.

Note

This rule doesn't affect the creation of combinations with Marketing cost center and company values other than Operations.

Rule Complexity

For optimal performance and ease of understanding, define several simple validation rules instead of using one complex rule. Simple validation rules let you provide a more specific error message and are easier to maintain over time.

Avoid rules that control validation across more than two segments, where possible. While you can define cross validation rules that span two or more segments, keep in mind that it becomes more difficult to interpret cross validation error messages and correct invalid key flexfield combinations as your rules encompass more segments.

Maintenance

To maintain consistent validation, review existing key flexfields when you update your cross validation rules. Regardless of your current validation rules, Oracle Fusion Applications accept a key flexfield combination if the combination already exists and is enabled. Therefore, to ensure accurate validation, you must review your existing combinations and disable any combinations that don't match the criteria of your new rules.

Tip

To keep this type of key flexfield maintenance to a minimum, decide upon your cross validation rules when you first set up your key flexfield structure. Define cross validation rules before creating combinations and before combinations are used in transactions.

If you want to prevent users from using previously existing combinations that are no longer valid according to your cross validation rules, disable those combinations using the combinations page.

Creating a Cross Validation Rule: Example

Create cross validation rules to prevent specific combinations of segment values in your account combinations, for example, preventing a particular cost center from being combined with a specific company value. Cross validation rules only affect the creation of new account combinations.

Scenario

Enter a new cross validation rule to prevent your InFusion America Inc. company value 01 from being combined with your marketing department value.
Define Applications Core Configuration

300 in an account combination. Your company, InFusion America Inc. does not have a marketing department.

1. Navigate to the Manage Cross-Validation Rules task from within your implementation project, and then click the Go to Task icon.
2. Select your InFusion America chart of accounts.
3. Click the Create icon.
4. Specify a unique rule Name, IFAM01, and an optional Description, Do not combine Marketing Department, 300 with InFusion America, company 01.
5. Enter an optional effective From Date of today. Check Enabled.
6. Click the Change filter condition on the Condition Filter. Enter Company equal to 01. The cross validation rule evaluates if Company 01 was entered and if it was entered, then the validation process continues to evaluate the rule.

Note
If you do not specify any statement in the condition filter, then the rule is always evaluated.

7. Click on the Change filter condition on the Validation Filter. Enter Cost Center is not equal to 300. When the rule is evaluated, an account combination must contain a cost center other than 300 before it can be created.
8. Enter an Error Message: Cost Center 300 is not allowed with Company 01. The message displays in the relevant user interfaces and processes when an account combination cannot be created because it violates the rule.
9. Click Save and Close.

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, use the Import Oracle Fusion Data Extensions for Transactional Business Intelligence process to import
the flexfield changes into the Oracle Business Intelligence repository. Users can make use of the newly-generated attributes in business intelligence applications. For additional information about logical objects and import, refer to the Oracle Transactional Business Intelligence Administrator's Guide.

**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 didn't assign a label, the flattened components include a discrete attribute for each BI-enabled segment instance in each structure.

**Mapping to Logical Objects in Business Intelligence**

You can simplify reporting by representing similar segments as a single logical object in Business Intelligence.

If you assign a label to segments that serve the same purpose in different structures, you can consolidate or equalize the segments into a single attribute. This prevents duplication and the extra workload and complexity that result from the flattening process. For example, an organization may have more than one definition of its key accounting flexfield to support different requirements for accounting reporting, or due to chart of accounts definitions from acquired organizations. A US Accounting Flexfield structure may have a segment called Subaccount to track project expenditures. The same type of information may be tracked in a UK accounting flexfield structure with a segment called Project. Equalize these two segments to create a single list of values for reporting.

Non-labeled segments aren’t equalized across context values, 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.

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 doesn’t 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.

---

**Importing to Oracle Business Intelligence Repository**

After you deploy a business intelligence-enabled flexfield, import the flexfield changes into the Oracle Business Intelligence repository to make use of the newly flattened business components in business intelligence and then propagate the flexfield object changes. When you import the metadata into the Oracle Business Intelligence repository, you must do so as the FUSION_APPS_BI_APPID user.

---

**Note**

To import flexfield changes into the Oracle Business Intelligence repository in Oracle Cloud implementations, run the Import Oracle Fusion Data Extensions for Transactional Business Intelligence process. For additional information about import, refer to the Oracle Transactional Business Intelligence Administrator’s Guide.

---

**Tip**

When you import a flexfield into the Oracle Business Intelligence repository, you see both `<name>_` and `<name>_c` attributes for each segment, along with some other optional attributes. The `<name>_` attribute contains the value. The `<name>_c` attribute contains the code of the value set that the value comes from, and is used for linking to the value dimension. You must import both attributes.

---

**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 Combinations Table for Entering Accounts and Employees

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

The figure shows the origin in the code combinations 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 combinations 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 combinations table.

If dynamic combination creation isn't 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 doesn't 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 is 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.
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 management repository provided by Oracle WebCenter Content Server. Users managing attachments have no real interaction with the repository unless the repository mode is enabled for attachments on specific business objects. In that case, users can share attachments among objects, update attachments by checking them out of and back into the repository, and perform other tasks. Access to attachment files is controlled by a digital signing mechanism. Depending on security, users might have direct access to the repository.

Security

Data security that applies to a specific business object also applies to attachments for that object, as determined by the attachment entity defined for the object. For example, if a user has no access to a specific expense report, then the same user cannot access attachments for the expense report. You can also use attachment categories to control access and actions on attachments, based on roles associated with the category. For more information on securing attachments, see the Oracle Fusion Applications Developer’s Guide.

Attachment Entities: Explained

An attachment entity is usually a database entity, for example a table or view, that represents a business object attachments can be associated with. Each attachment UI must be defined with a corresponding attachment entity, which not only identifies the business object to attach to, but also controls what users can do. Attachment entities are used only in the context of attachments and exist separately from the database entities that they are based on.

Edit and create attachment entities on the Manage Attachment Entities page, which you can access by starting in the Setup and Maintenance Overview page and searching for the Manage Attachment Entities task. Though you would
generally use predefined attachment entities with attachment UIs, you might need to create new entities, for example when developing custom UIs.

Entity Names

An attachment entity name should match the name of the table or view that represents the business object to attach to. The name is also used in the repository folder that is automatically created to store attachments for the entity. The attachment entity display name should be something that users know to represent the business object.

Database Resource

The data security policies associated with the database resource defined for the attachment entity would apply to attachments for that entity. For example, based on the database resource for the expense reports attachment entity, the same policies apply to attachments for expense reports. The database resource value must match the value in the OBJ_NAME column in the FND_OBJECTS table for the business object that the entity represents.

Enabling Security

Security based on the database resource associated with the attachment entity is always in effect. What you can enable or disable is security based on attachment categories. If any of the attachment categories associated with the attachment entity has data security defined, then that security applies to this entity only if enabled.

Attachment Entities and Attachment Categories: How They Work Together

The association between attachment entities and categories determines which categories can be used for an entity. For example, categories associated with the expense report attachment entity are available to be implemented in attachment UIs for expense reports. You can define these associations when managing either entities or categories. Any association changes in either the Manage Attachment Entities or Manage Attachment Categories page are reflected on the other page. You can access either page by starting in the Setup and Maintenance Overview page and searching for attachment tasks.

Managing Entities

You determine which attachment categories are relevant to a particular entity on the Manage Attachment Entities page, and each entity must have at least one category. Depending on configuration, any or all of the available categories for that entity are used. For example, you assign three categories to the expense reports attachment entity. For a particular expense report page with attachments functionality, you can customize the attachments component to specify which of the three categories are used. Based on your selection, the data security defined for each category, if any, is applied to attachments on that page if the attachment entity has category-based security enabled.
Managing Categories

If you create an attachment category and need to assign it to multiple attachment entities, use the Manage Attachment Categories page. The association means the same as the association on the Manage Attachment Entities page.

Attachments Troubleshooting: Explained

Attachments UIs for users to add and manage attachments are fully functional as is, and users usually would not encounter issues. If you customize attachments in any way, for example by creating additional attachment categories and implementing data security on them, then some issues might arise.

Issue: Unable to View, Add, Update, or Delete Attachments

Users encounter issues when trying to view attachments or perform actions such as adding attachments.

- Users can no longer see specific attachments that they were previously able to see.
- Likewise, they can no longer update or delete attachments.
- Users get an error stating that they do not have permission to add attachments.

Resolution

Use the Manage Attachment Entities page to ensure that attachment categories are associated to the relevant attachment entity. For example, if users can no longer see attachments for an expense report, then search for the expense report attachment entity and assign all necessary categories to it. You might need to check with your system administrator or help desk to determine the exact entity used on the page with the expenses attachments or what categories to assign.

If data security is implemented on the categories for the attachment entity, then verify that the Enable Security check box is selected in the Manage Attachment Entities page for that entity. Make sure that users have a role with the privileges shown in the following table, to view, add, update, or delete attachments with a specific attachment category.

<table>
<thead>
<tr>
<th>Action</th>
<th>Privilege</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>Read Application Attachment (FND_READ_APPLICATION_ATTACHMENT_DATA)</td>
</tr>
<tr>
<td>Add or Update</td>
<td>Update Application Attachment (FND_UPDATE_APPLICATION_ATTACHMENT_DATA)</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete Application Attachment (FND_DELETE_APPLICATION_ATTACHMENT_DATA)</td>
</tr>
</tbody>
</table>

For example, if users have the Read Application Attachment privilege for all categories associated with the expense report attachment entity, except the
Receipts attachment category, then they can view all expense report attachments except those created with the Receipts category. Likewise, if users do not have the Update Application Attachment privilege for any attachment categories tied to the expense report attachment entity, then they cannot create any attachments at all for expense reports.

For more information on attachment category data security, see the Oracle Fusion Applications Developer’s Guide.

Finally, certain attachments UI for users have predefined restrictions on categories in place. Your developers can also introduce additional filters to determine which document categories are available for a specific page. Check with your developers or help desk.

**Issue: Missing Attachment Category**

Users can see existing attachments, but the attachments no longer have an attachment category value.

**Resolution**

When the attachment was added, at least one category existed for the corresponding attachment entity, as otherwise the attachment could not have been added. Since then, the entity was edited so that it no longer has any assigned categories, so the user cannot see the category associated with that attachment.

Use the Manage Attachment Entities page to reassign attachment categories to the relevant attachment entity. For example, if users can no longer see the Receipts attachment category for an attachment to an expense report, then search for the expense report attachment entity and assign to it the Receipts category. You might need to check with your system administrator or help desk to determine the exact entity used on the page with the expenses attachments or what additional categories to assign.

Finally, certain attachments UI for users have predefined restrictions on categories in place. Your developers can also introduce additional filters to determine which document categories are available for a specific page. Check with your developers or help desk.

**FAQs for Define Attachments**

**What’s an attachment category?**

An attachment category is used to classify and secure attachments. Each attachment user interface must be defined with at least one category for users to be able to add attachments. If there are multiple categories, users can view them and select one when adding attachments. For example, attachments for an expense report can be categorized as receipts, scanned invoice images, and so on.

You can also associate roles with categories to determine user access and actions for attachments, based on the categories assigned to the attachment entity. For
example, security for expense report attachments can be based in part on the
categories assigned to the expense report attachment entity. You can define
multiple categories per module, and add and manage custom categories for your
own purposes. For more information on attachment category data security, see
the Oracle Fusion Applications Developer's Guide.

Use the Manage Attachment Categories page, which you can access by starting
in the Setup and Maintenance Overview page and searching for the Manage
Attachment Categories task.
Define Product Information Management Configuration

Product Information Management Features by License: Overview

Product Information Management is comprised of two products.

- Oracle Fusion Product and Catalog Management
- Oracle Fusion Product Hub

<table>
<thead>
<tr>
<th>Product License</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Fusion Product and Catalog Management</td>
<td>Define Basic Items</td>
</tr>
<tr>
<td></td>
<td>Define Basic Catalogs</td>
</tr>
<tr>
<td></td>
<td>Define Basic Product Structures</td>
</tr>
<tr>
<td></td>
<td>Define Advanced Items (also will get Define Basic Items by dependency in ADR)</td>
</tr>
<tr>
<td></td>
<td>Define Change Orders</td>
</tr>
<tr>
<td></td>
<td>Define Product Rules</td>
</tr>
<tr>
<td></td>
<td>Define Advanced Product Structures (also will get Define Basic Product Structures by dependency in ADR)</td>
</tr>
<tr>
<td></td>
<td>Manage Advanced Item Profile Options</td>
</tr>
<tr>
<td></td>
<td>Define New Item Requests</td>
</tr>
<tr>
<td>Oracle Fusion Product Hub</td>
<td>Define Product Source System</td>
</tr>
<tr>
<td></td>
<td>Define Advanced Catalogs (also get Define Basic Catalogs by dependency in ADR)</td>
</tr>
<tr>
<td></td>
<td>Define Data Quality for Products</td>
</tr>
<tr>
<td></td>
<td>Manage Advanced Item Profile Options</td>
</tr>
</tbody>
</table>
Define Basic Items

Product Value Sets: Explained

Value sets are specific to the application in which they will be used. In the Oracle Product Information Management application, value sets are used primarily for defining attributes where the values that an attribute can have is limited to a specific set of values.

Value sets can be edited or new value sets can be created from the Manage Product Value Sets page. The Edit icon launches the Edit Value Sets page, which redraws in the same region of the local area. The Create icon launches the Create Value Sets page, which redraws in the same region of the local area.

A value set is defined by the value set code and is specific to the module of an application in which the value set is to be used, such as Item Class.

The validation type determines how the value of field is validated for the assigned value set. The following are the seeded values:

- Format Only
- Independent
- Dependent
- Subset
- Table

The value data type determines the data type for the value set. The following are the seeded values:

- Character
- Number
- Date
- Date/Time

Product Child Value Sets: Explained

The Manage Product Child Value Sets task uses the same page as the Manage Product Value Set task.

A child value set is used to define variants for stock-keeping units or SKUs. A SKU contains the common properties for an item. For example, a shirt can be produced with colors; white, red, yellow, and blue. The variant is used to represent the colors of the shirt.

You define child value sets as follows:

- Create a value set with validation type of independent, for example All Colors.
- Select the new value set in the Manage Product Value Sets results table, for example All Colors.
- Click Manage Values, create several values, for example Blue, Red, Green, Yellow, and Black.
• Create a value set with validation type of Subset and enter the first value set you created for the independent value set, for example: Summer Colors.

• Select the value set Summer Colors in the Manage Product Value Set result table.

• Click Manage Values and then click the Add icon. The dialog will show a list of values based on the value set named Summer colors. Select two of them.

The value set Summer Colors is a child of All Colors.

Managing Default Item Class: Explained

The Root Item Class is seeded and all item classes are created as children of the Root Item Class. For Oracle Fusion Product Model customers, only the Root Item Class is available. The Manage Default Item Class task enables Product Model customers to manage item class templates, descriptive flexfields, attachment categories and lifecycle phases. The Manage Default Item Class task launches an edit page for the Root Item Class.

The functionality for the Root Item Class is defined using three tabs:

• The Basic tab enables descriptive flexfields and attachment categories to be viewed and managed for the Root Item Class.

• The Lifecycle Phases tab enables one or more lifecycle phases of a lifecycle to be associated with an item class.

• The Templates tab is where you define and manage item templates for the item class.

Item Statuses: Explained

In the Item Status table, select a status code to display the associated attribute groups and attributes as well as control information.

You can create or edit or delete item statuses on the Manage Item Statuses page. Inactive dates are used to specify the date after which the item status will no longer be active. Operational attribute groups and attributes corresponding to the selected item status are displayed in the Details section. Select a value for the status from choice list for the attribute. Whenever the status is applied to the item, the value of the attribute may change. If the status will have no value, select No.

Select the Usage value of None or Defaulted or Inherited in the choice list for the Usage field that corresponds to how the attribute value will change based on the item status value:

• Defaulted- Sets the values of the item status attributes when the status value changes, but allows the overriding of the value during import and update of item.

• Inherited- Sets the values of the item status attributes when the status value changes, but overrides cannot occur.

• None- The item status attribute values will not be changed.
Any change made to an item status is not applied automatically to existing items, but will be applied during the editing of an item when the item status value is changed.

Managing Item Types: Explained

Item types are managed on the Manage Item Types page. There are 32 seeded item types and you can edit them or create additional item types. Item types are date-enabled and are made active or inactive by adjusting the Start Date and End Date. To benefit from the use of item types, you must enable them by selecting the Enable checkbox.

Cross-Reference Types: Explained

Cross-References provide the functionality to map additional information about an item in the form of a value and cross-reference type. For example, the cross-reference can map between an item and an old part number, where the value is the value for the old part number and the type is Old Part Number. Cross-Reference Types are part of item relationships where the item relationship type is Cross-Reference. There are no values seeded for cross-reference types. You define the values using the Manage Cross Reference Types task. Cross-reference types are date-enabled and can be made active or inactive by adjusting the values of the Start Date and End Date. To benefit from using item relationship for cross-reference, you must enable cross-reference types by checking the Enable checkbox.

Item Imports: Explained

You can import items and item-related information using interface tables. This import data is loaded into the production tables using the Import Item task.

Import Item

The Import Item task creates an Enterprise Storage Server (ESS) process that takes the data that is loaded in the interface tables and uses the import process to move the data to the production tables. The import processes will perform all of the validations necessary to ensure the data imported is correct prior to moving the data into the production tables.

1. Access the Enterprise Storage Server and provide a process name (job definition) such as Item Import Process.
2. Select Setup and Maintenance from the Navigator. Access the All Tasks tab on the Overview page, and search for the Import Item task with the name of your ESS process definition.
3. Click the Go to Task icon in the search results for that Import Item task.

The parameters for the item import process are

- **Batch ID**: Associate the interface table to an item batch definition.
- **Organization**: Select an organization to be used for the import.
- **Process Only**: Determines how the data is processed. The choices are:
  - Create
  - Sync
  - Update
- **Process All Organizations**: Select Yes if the import contains items that will be imported to multiple organizations.
- **Delete Processed Rows**: Select Yes to delete rows that are imported without errors

4. Click Submit and the Request Number will be displayed.

**Monitoring Import Items**

Select Setup and Maintenance from the Navigator, then access Monitor Item Imports to search for specific Enterprise Storage Server processes and monitor their status in the search results table.

**Related Item Subtypes: Explained**

A related item is an item relationship between two existing items. How the two items are related is defined by a subtype. Multiple subtypes for related items are seeded, and you define additional subtypes on the Manage Related Item Subtypes page.

**Managing Descriptive Flexfields for Items: Explained**

You can use descriptive flexfields to capture additional information about items beyond what is provided by the predefined set of operational attributes in Oracle Fusion Product Model.

**Item Descriptive Flexfields**

If you are not using Oracle Fusion Product and Catalog Management, then you cannot create user-defined attribute groups and attributes. However you can use descriptive flexfields associated at Item level to create fields to capture information about items. Like other descriptive flexfields, item descriptive flexfields have context segments and context-sensitive segments whose values are validated on entry by value sets. You can define the value sets to control...
what values users can enter in a descriptive flexfield segment. Examples of information that you might capture are size and volumetric weight.

Manage this flexfield type by using the Manage Item Descriptive Flexfields task, which you can access by searching for flexfield tasks on the Setup and Maintenance Overview page.

**Item Revision Descriptive Flexfields**

Use descriptive flexfields associated at Item Revision level to capture item revision information whose values may differ between revisions of the same item.

Manage this flexfield type by using the Manage Item Revision Descriptive Flexfields task, which you can access by searching for flexfield tasks on the Setup and Maintenance Overview page.

**Item Relationship Descriptive Flexfields**

When defining descriptive flexfields associated with item relationships, you must use certain prefixes when naming the context segments, in order for the segments to be displayed for the respective relationships.

The prefixes required for naming the context segments are listed in the following table, with their corresponding item relationship types. For example, if you define an item relationship descriptive flexfield with a context segment named `RELATED_RELATIONSHIP_ATTRIBUTES`, then the value segments of this context will be displayed for Related Item Relationships when users conduct transactions in that context. For another example, when users navigate to a UI of a particular object, such as a Competitor Item, they see the contexts whose internal name has the prefix `COMP`.

<table>
<thead>
<tr>
<th>Relationship Type</th>
<th>Prefix for Context Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Competitor Item Relationship</td>
<td>COMP</td>
</tr>
<tr>
<td>Customer Item Relationship</td>
<td>CUST</td>
</tr>
<tr>
<td>Item Cross-reference Relationship</td>
<td>XREF</td>
</tr>
<tr>
<td>GTIN Relationship</td>
<td>GTIN</td>
</tr>
<tr>
<td>Manufacturer Part Number Relationship</td>
<td>MFG</td>
</tr>
<tr>
<td>Related Item Relationship</td>
<td>RELATED</td>
</tr>
<tr>
<td>Source System Item Relationship</td>
<td>SYS</td>
</tr>
</tbody>
</table>

Manage this flexfield type by using the Manage Item Relationship Descriptive Flexfields task, which you can access by searching for flexfield tasks on the Setup and Maintenance Overview page.

**Trading Partner Item Descriptive Flexfields**

When defining descriptive flexfields associated with trading partner items, you must use certain prefixes when naming the context segments, in order for the segments to be displayed for the respective trading partner type.
Define Product Information Management Configuration

The prefixes required for naming the context segments are listed in the following table, with their corresponding trading partner types. For example, if you define a trading partner item descriptive flexfield with a context segment named COMP_TPI_ATTRIBUTES, then the value segments of this context will be displayed for Competitor Item when users conduct transactions in that context.

<table>
<thead>
<tr>
<th>Trading Partner Type</th>
<th>Prefix for Context Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Competitor Item</td>
<td>COMP</td>
</tr>
<tr>
<td>Customer Item</td>
<td>CUST</td>
</tr>
<tr>
<td>Manufacturer Item</td>
<td>MFG</td>
</tr>
</tbody>
</table>

Manage this flexfield type by using the Manage Trading Partner Item Descriptive Flexfields task, which you can access by searching for flexfield tasks on the Setup and Maintenance Overview page.

FAQs for Define Basic Items

**What's lifecycle validation?**

Lifecycle validation enforces compatible lifecycle phases between parent and component items in an item structure, at the structure name level. For example, an item in the Design lifecycle phase can only have component items that are in the same lifecycle phase or the next phase. The only exception is for items in the Obsolete lifecycle phase, because this is considered the end of life for an item. Items in the Obsolete lifecycle phase cannot be added to a new structure.

The lifecycle phases for a computer component lifecycle might be:

- Concept
- Design
- Prototype
- Pre-Production
- Retirement (Obsolete)

**What's the difference between lifecycle phase types and lifecycle phases?**

Lifecycle phase types are seeded and describe the type of lifecycle phase. They are Design, Obsolete, Preproduction or Prototype, and Production.

Lifecycle phases must be created by the user by selecting one of the seeded lifecycle phase types.
Define Basic Product Structure

Product Value Sets: Explained

Value sets are specific to the application in which they will be used. In the Oracle Product Information Management application, value sets are used primarily for defining attributes where the values that an attribute can have is limited to a specific set of values.

Value sets can be edited or new value sets can be created from the Manage Product Value Sets page. The Edit icon launches the Edit Value Sets page, which redraws in the same region of the local area. The Create icon launches the Create Value Sets page, which redraws in the same region of the local area.

A value set is defined by the value set code and is specific to the module of an application in which the value set is to be used, such as Item Class.

The validation type determines how the value of field is validated for the assigned value set. The following are the seeded values:

- Format Only
- Independent
- Dependent
- Subset
- Table

The value data type determines the data type for the value set. The following are the seeded values:

- Character
- Number
- Date
- Date/Time

Item Structure Type Details: Explained

Item structure type is a seeded value that enables you to define the composition of an item. You can add additional structure names and specify item attachment categories that need to be available when viewing or managing structures. Item structure types provide a common set of metadata to all structures defined under it.

Structure Type Details

A structure type enables you to classify and characterize item structures throughout the lifecycle of an item. You can define an unlimited number of structure names for a structure type, enabling you to create several different structures of the Item Structure Type that you can use for comparison.

A seeded structure name (Primary) is available. Only the description and lifecycle validation flag can be updated for this structure name. It will always be active.
You can associate item attachment categories with a structure type.

**Define Basic Catalogs**

**Catalogs: How They Work Together**

A catalog is a collection of categories that you use to classify items. You can organize the categories into a hierarchy that represents a taxonomy. You create new categories only in the context of a catalog. You can add existing categories to one or more catalogs, either from another catalog or as shared categories from a source catalog.

You can set the **Catalog Content** value to **Items at all levels** which allows items to be assigned to any level within the category hierarchy, not only to the leaf levels.

The following diagram shows the relationships of the catalog components.
first level in the category hierarchy are associated with the catalog root through the catalog category association component.

Category

A category is a component of a catalog that represents a portion of the classification defined by the categories and category hierarchy in the catalog. You can associate a category to a catalog through the catalog category association. Both the shared category and the native category are associated through the catalog category association.

Catalog Category Association

Catalog category association represents the relationship between a catalog and a category, or a parent category and a child category. Each catalog category association represents one relationship between the catalog and a category or one relationship between a parent category and a child category.

Item Category Assignment

Item category assignment represents the assignment of the item to a category in a catalog. Each item category assignment represents the relationship between a category and an item.

Item

An item represents objects such as a product, service or template. An item is assigned through the item category assignment component.

Attachment or Image

Information is associated to the catalog and/or category, or both, through the attachment framework. Multiple attachments are supported but you can associate only a single attachment or attachment type image with a catalog or category.

Items to Categories Assignment: Explained

You can assign items to categories on the Edit Catalog page, category hierarchy tab, on the category detail item tab. You can assign items only to active categories and categories where the Category Content field value is Items and Categories or Items Only. In addition, you can configure catalogs to control item assignment to categories within the catalog by selecting the Allow multiple item category assignment check box, which allows items to be added to all levels of the category hierarchy.

You select items from a choice list and add them to the category. The choice list is filtered based on a set of rules:

- Item data level security: Displays only the items that the user has permission to view and assign.
- Organization context: Based on the organization context that is controlled by a choice list in the item table header, only the items assigned to organizations are displayed.
Controlling Item Assignment

You also control item assignment by selecting the value of the Controlled at check box. If you select the Master Level value and the organization context is a master organization, the items are automatically assigned to all child organizations that are associated with the master organization.

Catalog Publishing: Explained

Other applications can use catalog data if you export the catalog content. For example, you may want to export catalog content to use as a monthly report of all items assigned to a specific catalog. You can use the default publish template provided in hyper text markup language (HTML). You can specify the content and layout of the catalog information. When the catalog is published, you select the format and initiate the creation of the content in the file.

The following aspects are important regarding catalog data to be published:

- Publish a catalog
- Type of catalog content that can be published

Publish a Catalog

You initiate a search for a catalog from the Manage Catalogs page, select the row corresponding to the catalog that you want to publish and select the Publish action. The application generates the report based on the default template in HTML format, and the locale prior to creation of the file. You can select a new template or format from the report window. The content displayed for items, categories, catalog categories, and catalog is based on the publish template.

Type of Catalog Content That Can Be Published

The default catalog publish template allows the publication of the catalog header details, category hierarchy, category details, and category item assignments. The order of a published report begins with the catalog header and the catalog category details. If the category has a child relationship then the catalog category association details for the child category follows. If the child category has a hierarchy, then the complete hierarchy under the category is published with the catalog category association details and categories details.

FAQs for Define Basic Catalogs

How can I share catalog content?

Categories can be shared across multiple catalogs allowing catalog content to be reused and saving the work needed to maintain multiple copies of the categories. In the case of category sharing, the category structure in the source catalog can be different than the native catalog.
Categories can be shared using two methods; the first method is directly associating the category to the catalog. The category is added to the catalog and can be edited in the catalog or any catalog the category is associated to. The items assigned to the category are not shared, but are assigned to the category in context with the catalog the category is associated. For example if the category name or description is changed in one catalog, the change will be reflected in all catalogs where the category is associated, but if items are assigned to a category, the assignment will be for that single catalog.

The second method of sharing categories is adding a category by reference into the catalog. During the creation of the catalog, sharing can be enabled by specifying a single source catalog that will be used for sharing by reference and setting the value of the sharing content to control what content will be shared from the source catalog. The advantage of using sharing by reference is source catalog content can be shared to multiple catalogs and maintained in a single place, the source catalog. In addition, the referenced content can be more than one category, for example a complete category hierarchy and any assigned items to categories in shared content can also be reference within the catalog.

Define Advanced Items

Item Classes: Explained

When you are ready to create or edit an item class, you must decide whether to allow items to be created under the item class.

To create an item class, perform the following steps:

- Create a list of all items.
- Classify or categorize these items (item classes).
- Define any parent-child relationships (item class hierarchy).
- Gather the unique types of specifications required for each type of classification at a high level (user-defined attribute groups).
- Gather the unique specifications required within the group (user-defined attributes).
- If there are specified values that must be used, define them (value sets and values).

Item classes can be used for classification purposes and in some case, item creation may not be allowed. By optionally setting the Item Creation Allowed attribute to No, item creation under a item class can be prevented. However, a child item class of such an item class may be allowed for item creation. For example:
This prevents items from being created in Computers and Desktops, but allows items to be created for Green Desktops and Gaming Desktops. Optionally, specify a date on which the item class will become inactive. You cannot specify an inactive date that is later than the inactive date of an item class parent, nor can you specify an inactive date that has already passed. Also, all children of a parent item class with an inactive date should be made inactive at the same time or earlier.

Attachment categories enable you to categorize and classify attachments to an item. To classify item attachments, associate attachment categories with item catalog categories. Associated attachment categories are inherited down through the item class hierarchy.

Version control allows new versions to be created for all items in an item class. An integrated workflow definition permits the creation of custom new item request definitions. This workflow definition establishes a multistep process for routing the definition and approval of items. When a new item is created, various people in the organization use the workflow to define aspects of the item, like base operational attributes, user-defined attributes, structures, attachments, categories associated, and organization assignments.

**Note**

Versioning an item and creating a new item request are independent of each other. Versions can be added to a new item request and routed for approval if there is a business need.

When setting up definition steps for a new item request at the item class, you can identify various item details as mandatory, at each step. Definition of entire entity can be made mandatory or just certain attributes. This ensures that the item information required for a downstream step is defined and available for use.

Required attributes can be inherited from parent and assignee access is validated.

You can control item creation, viewing and update access by assigning a role on the item class to a principal or group of users. Security allows a person or a group to have privileges to an item of item class in each organization. This is inherited and hence a person who has a privilege in a parent item class will automatically have the same privilege in the child item classes.

**Transactional Attributes: Explained**

Attributes that exist for each instance of an item and the values for the attributes can be different.

For example:

- The number of megabytes (MB) or gigabytes (GB) of e-mail storage on a digital subscriber line (DSL) account.
- The monogram text on a shirt pocket.
- The color of a music player.
These attributes are defined at the item class and their attribute value is captured at the time of a transaction by downstream applications. The metadata values of these attributes are maintained at the item class. Order orchestration and order capture systems are two examples of downstream use. All transactional attributes must be associated with a value set.

The following metadata values can be defined for an attribute.

- **Required**: Indicates whether the attribute value is required at the transaction.
- **Default Value**: Indicates the default value of the attribute.
- **Value Set**: Indicates the value set associated with the attribute.
- **Read Only**: Indicates whether the attribute value is read only.
- **Hidden**: Indicates whether the attribute is not shown.
- **Active**: Indicates whether the attribute is active or inactive.

Transactional attributes are inherited across the item class hierarchy. The metadata is data-effective. Changes in the metadata will be reflected immediately at the item level. For example:

- Any of the metadata of a transactional item attribute belonging to a specific domain, if modified in the child item class would break the inheritance. Any changes done at the parent item class for this transactional item attribute would not get inherited. Multiple records with same date range can exist if they belong to different domains. For example, the transactional item attribute Memory is associated with a Domain and order capture. Each of the domains may use a different set of metadata for its own purpose. Hence, for the same date range, two different records can exist. Only Start Dates for a transactional item attribute would be entered by a user. End date would be calculated automatically based on the next Date Effective record.

- Users can modify (either Start Date and metadata) of a future effective record. Records with Starting date as Past cannot be modify or edited.

- Only start dates can be set to permit updating by a user, and the end date of a record will automatically be pulled from the next record.

- Any changes performed in the parent item class would be inherited by the child item class. If the corresponding record is modified in the child, then these changes will not be inherited.

Item pages provide a mechanism with which to customize the user interface.

**Pages and Attribute Groups**

Pages and attribute groups enable you to structure your data.

Benefits include:

- You can combine and sequence attribute groups into pages.
- There is no limit on the number of attribute groups associated with a page.
- Pages can be created at item class and are inherited down the item class hierarchy.
• Attribute groups can be added to pages sequentially and based on this sequence, these attribute groups are shown in items.

• Attributes groups can be added for an inherited page at the child item class.

Functional Item pages are another type of special pages which are used to associate pages already created for use in the application. Application scope indicates the application which uses these pages and the usage indicates the specific use of the configured pages.

Data Quality

You can associate attributes for the purpose of standardization and matching, to be performed when items are created. You restrict the attributes to be processed for standardization or matching or both. Selecting Standardization allows the data quality engine to return the standardized values for these attributes. Matching allows the data quality engine to return any existing items which matches the value of these attributes and are potential duplicates.

Lifecycle Phases

Sequential lifecycles phases enable you to track and control the lifecycle phases of items. Each phase represents a set of tasks and deliverables that are required before promoting the item to the next phase. You can associate lifecycle phases to an item class which are created elsewhere. Lifecycle phases are inherited down the item class hierarchy and new lifecycle phases can be added to child item classes. For example, the lifecycle phases for a computer component item class might be: Concept, Prototype, Production, and Retirement.

Templates

Template is a defined set of attribute values used during item creation. When you apply a template to an item, you overlay or default-in the set of attribute values to the item definition. For example, every time users in a particular organization create new items, the attributes, as defined and approved by the organization appear in the appropriate fields. No user guesswork is required, and time is saved during the creation of items with a similar form, fit and function. Templates are created for each item class. Templates are specific to organization. Templates are inherited down the item class hierarchy. You can define both operational attributes and user defined attributes for each template.

Search and Display Format

Search formats provide a convenient way to save frequently used search criteria. Search formats created at item class will be available to all users. Search formats are always created in the context of item class. Display formats enable you to predefined search display views. You can use these views to look at different sets of item attributes that are returned by the search. Display formats created at item class will be available to all users. Display formats are always created in the context of item class.

Import Format

An import format identifies the base and user-defined attributes in an item class that are imported into the application using a spreadsheet. Consequently, when
you import item business entities from a spreadsheet, the items are all imported into the particular item class defined in the import format. These imported item business entities inherit all the attribute groups defined for the specific item class. You cannot edit the layout of an import format once it is created.

FAQs for Define Advanced Items

How can I create an item class?

To create an item class, select a parent item class on the Item Class Search Results page and select Create. Provide the required information, and optionally include additional details, such as attribute groups, pages, templates, and search and display formats.

Define Change Orders

Change Order Definitions: Explained

The following components are required for creating change orders:

Change Priorities and Change Reasons

Change priorities are used to identify the criticality of a change while change reasons are used to categorize and establish the cause or reason for a change.

Processes and rules can be defined around change priorities and change reasons so that change orders with a high priority or with safety related reasons get priority over other change orders or they get routed through different levels of approvals.

Predefined values for change priorities and change reasons:

- Priorities: High, Medium, and Low
- Reasons: Cost, Quality, and Safety

Additional values for change priorities and change reasons can be added by accessing the task’s Manage Change Reasons and Manage Change Priorities in Functional Setup Manager.

Change Statuses and Status Types

Status indicates a stage in a change order’s lifecycle. You can create new statuses within predefined status types.

Predefined values for status type:

- Open, Interim Approval, Approval, Scheduled and Completed
New statuses can be added and managed by accessing the Manage Change Statuses task in Functional Setup Manager.

**Change Order Types**

Each change order is created within a change order type. Change order types contain the following information:

- Number generation method
- Workflow
- Propagation rules

**Define Product Rules**

**Rules and Rule Sets: Explained**

Rules define integrity constraints on the attributes of items and structures. Define integrity constraints on operational as well as on user-defined attributes. Integrity constraints often implement business rules and are created through use of the rules framework. For example, minimum speed must be less than maximum speed.

Rule sets gather multiple rules together and are assigned to an association type, such as attribute groups, item classes, change types, or structure types. They also list valid business entities. For example, item, item supplier, and item revision. This enables the expressions entered to be validated by checking for allowable entities.

If the rule set is assigned to an attribute group, then only the attributes in that group can be used in the rule.

If the rule set is assigned to an item class, then only the attribute groups valid for that item class will be used.

The status of a rule set can be identified as draft. Users can keep a rule set in draft status until the drafting of rules is complete. If so, the rule set is not triggered as regular transactions are completed. During this time, simulation analysis can be run to study the impact of the rule sets on a selected set of existing items, enabling users to make necessary changes. While performing the simulation, the draft rule sets along with other active rule sets are applied on the selected set of items, and the impact is captured; this is handled by an asynchronous scheduled process.

The rules for new item request and change order type associations are used to generate new item request and change order numbers.

There are three types of rule sets and rules:

- Validation
- Assignment
• Composite

**Validation**

Validation rule sets validate conditions based on attribute values defined for items. They are typically used to model predefined business rules on items.

Each attribute is referenced by its business entity and attribute group name followed by the attribute name. For example, item.Physical_Attributes.NetWeight. In this example, item indicates that it is an item attribute; Physical_Attributes is the internal name of the attribute group, and NetWeight is the internal name of the attribute.

Validation rules restrict items that can be added as related items to an item, and restrict the relationship types that can be allowed for items. This restriction could be based on item or item revision-level attributes which could be operational or EFF.

Test the validations by going to the Item Update screens and editing the appropriate attribute groups. Updated values are validated against the rules, and error messages appear on the screen.

**Assignment**

An assignment rule set determines the value of an item attribute based on the specified condition. Generally, rule sets for assignments should be executed before validations, since they permit you to write validation conditions ensuring that the result of the assignments are valid.

On the Edit Rule Set page, create a rule. For example, Lead Percent is Total Lead Mass divided by Unit Weight.

After you create a rule, validate and save it. This enters the first assignment rule into the rule set. Then, if necessary, enter subsequent rules. Rules are executed in the order of their sequence in the rule set. Therefore, if an attribute’s expression depends on a previously calculated value, you must ensure that the previous value appears ahead of the attribute and is therefore computed first.

**Composite**

Composite rule set can contain both validation and assignment rule sets. They can be used to aggregate rule sets that operate on different attribute groups and item classes.

You create a composite rule set on the Manage Rule Sets page. To define a composite rule set of mixed type, ensure it contains both validation and assignment rule sets. Set the type to Mixed, enabling the creation of a rule set that contains both assignment and validation rule sets. Then add assignment and validation rule sets to the composite rule set.

**What’s rule set assigning?**

Rules are not active until they are assigned to an appropriate business entity. You use the Setup Rule Set Assignments link to assign a rule set to either an item, item revision, or item supplier business entity.
What's a return type for?

Return type can be generic, in which case you can specify the target attribute group and target attribute for the value to be assigned to. Otherwise return type can be item description and item number to generate assignment rules.

What's a rule sequence?

Rule sequence dictates the order in which rules are executed. They are typically entered in 10s, but can be 1, 2, 3 or 100, 200, 300.

What's a rule-builder expression?

Rule-builder expressions enable you to specify the body of a rule. For example, an If Expression or a Validation Condition. Note that right-click functionality is enabled to assist in selecting attributes that participate in the rule expression.

What's a severity equals warning?

Severity can be warning, reject, or needs approval

In the case of warning, a message is displayed to the user but they are allowed to continue.

In the case of reject, a message is displayed and the user must fix the issue before proceeding.

In the case of needs approval, changes made by the user must be approved through a change order.

What's a validation condition?

A validation condition is evaluated when the result of an "If Expression" is "True." Only when the validation condition evaluates to "False" will a message be displayed. For example, if you created a rule where the width of an item is greater than its length and the validation condition is false, if a user tried to define an item where the width of the item is greater, a message will display saying the length must be greater than the width.
**What’s the difference between assignment rule sets and validation rule sets?**

Assignment rule sets are used to determine the value of an item attribute based on a specified condition.

Validation rule sets specify the rules to determine the validity of an item.

**What’s the hierarchy of rules and rule sets?**

Both assignment and validation type rule sets can be added to a composite rule set thus creating a hierarchy.

**What’s the user message and where does it display?**

The user message is displayed when the validation condition evaluates to FALSE. They can be displayed by the module consuming the rules framework and also, be displayed as a dialog box.

**Define New Item Requests**

**Number Generation Methods: Explained**

There are three number generation methods.

The three number generation methods are:

- Sequence Generated
- User-Defined
- Rules

**Sequence Generated**

If the new item request numbering method is sequence generated, then all item number requests will be automatically generated based on the sequence generation rules specified. You can specify a prefix, starting number, increment, and suffix. Enter a starting number and an increment value to determine the sequencing.
User-Defined

You enter the new item request number at the time of creation.

Rules

New item request number is generated based on the rule set associated with it.

New Item Request Definition Phase: Explained

During the definition phase, definition notifications are routed to participants for them to complete the item’s definition tasks.

Define an Item

From the Manage New Item Requests page, search for the New Item Request.

Launch the new item request from the Search Results table by clicking on a new item request link. That will take you to the New Item Request page.

Click on the Details sub tab and select the definition row. In the definition Workflow Details table, select Expand All from the Actions menu.

Click the Go to Task icon for the entity that needs definition. The item page appears where you enter the item information. The item definition page is rendered dynamically for each participant, where you can enter only the attributes or item entities for which you are responsible.

Note

You can also quickly navigate and drill down to an item detail page directly from a new item request definition and approval notification. A read-only item page will open a separate browser window.

You can identify various item details as mandatory at each step. This will ensure that item information required for a downstream step is defined and available for use.

Any item definition that is available to be set up for definition through new item request, can be identified as mandatory while completing the definition steps at the item class, including:

- Operational
- Extensible flexfield attributes
- Structures
- Packs
- Supplier associations
- Organization assignments
- Attachments
• Catalog category assignments
• Relationships

New Item Requests Workflow Processes: Explained

There are four predefined New Item Request (NIR) workflow statuses which enable you to perform various tasks when a new item request is created. You cannot create new statuses or delete statuses.

The four statuses are:

• Open
• Definition
• Approval
• Completed

Open

New item request attributes and items are defined and updated when the workflow status is open. A seeded request comment notification is associated with this status and the new item request cannot move to the next status until this notification has been responded to. The notification will be sent to the assignee and the requester of the new item request. However, to bypass the notification, assignees can be removed through customization by a programmer. Assignees would then need to promote the new item request to the next status manually or set it to automatically promote to the next status in the Business Process Execution Language (BPEL) process.

Definition

In the definition workflow status, you define the item information, such as specifications, structures, and packs. Additional items cannot be added in this workflow status. The definition workflow steps are defined at the item class level. Based on the definition steps, a notification is sent to those responsible for defining the item. Once the item is defined, you then promote the new item request to approval status manually or to be automatically approved in the BPEL process.

Approval

When approval is granted, the new item request cannot be modified. Approval notifications are sent to the approvers based on the rules set up in Approval Management Extensions (AMX). Once the new item request is approved, the status is automatically changed to completed.

Completed

During the completed phase, the NIR is in its final stage and its lifecycle is completed. It cannot be modified.
New Item Request Approval Phase: Explained

During the approval phase of a new item request, approval notifications are routed to the participants.

Approval notifications can be addressed to individual users or to multiple users in a predefined user group. From the notification, users can approve or reject the new item request.

Seeded new item request types enable you to configure automatic promotion and demotion rules at the new item request type level, allowing a new item request to move from one status to another without user intervention.

New item approval rules are defined and managed through the Approval Management Engine. For information on how approval rules are defined and managed, refer to the Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite.

FAQs for New Item Requests and Change Orders

What number generation methods are available?

There are three number generation methods available. Sequence Generated, User-Defined, and Rules.

Can I mark item definitions as mandatory?

Yes. When setting up the definition steps for New Item Requests at the Item Class, you will be able to mark various item details as mandatory, at each step. This ensures that item information required for a downstream step is defined and available for use.

Any item definition available to be set up for definition through new item requests can be marked as mandatory while creating the new item request definition steps at the item class. This includes entities such as:

- operational
- extensible flexfields
- structures and packs
- supplier associations
- organization assignments
• attachments
• catalog category assignments
• relationships

**Can I enforce New Item Request mandatory steps on the user side?**

Yes. Enforcing mandatory definition on the user side ensures that item definitions are completed as required by the item class. Any definitions marked as mandatory in the definition workflow at the item class level are enforced. Mandatory definitions will be identified in the definitions page and must be completed before allowing the workflow to proceed to the next step.

Missing definitions are listed during save and will cause an error message when clicking the done button on the definition notification.

**Can new item requests be automatically promoted and demoted?**

Yes. The seeded New Item Request Type provides promotion and demotion configuration. This enables you to set up automatic promotion and demotion rules at the new item request type level.

**Can Request Comment notification be skipped in a new item request in open status?**

Yes. When Skip Request Comment is enabled for open status, Request Comment notifications will not be sent for a New Item Request.

**Can I make changes to item organization assignment and route it for approval through a change order?**

Yes. You can enable an item in a child organization through a change order. By providing an effective date for the assignment, you can ensure that the item is available for transaction in the child organization from the specified date only. If approval routings are enabled for the change order, you can seek approvals on the item organization assignment before enabling the item in a child organization.

If approval routings are enabled for the change order, you can seek approvals on the item organization assignment before enabling the item in a child organization.
Can Request Comment notification be skipped in a change order in open status?

Yes. When Skip Request Comment is enabled for open status, Request Comment notifications will not be sent for a change order.

How can I view the mandatory definitions when defining an item, viewing notifications or from new item request details page?

When defining an item, below the header View Required Definitions link will open a window listing the required definition entities. From notification and New Item Request Details page click the Details icon in the Required Definition Values column which will open up the same listings.

What does All Values Required mean?

When checked definition of the entire entity is required and business entity is Item Operational Attributes and All Values Required is checked, all attributes within the selected attribute group are moved by default to selected list and the view is read-only.

Can mandatory definitions be inherited?

If the association is inherited and All Values Required is not checked, then inherited-required attributes from parent will be displayed in read only format under selected list and you can move additional required attributes from available list to selected list.

If the association is inherited and All Values Required is checked in parent, then All Values Required is read only and inherited attributes will be placed under selected list also in read only format.

Define Product Source Systems

Product Spoke Systems: Explained

Manage system attributes for the import and data quality options of the data being consolidated and imported into Oracle Product Information Management Data Hub.
Spoke System Setup

The set up of a spoke system is accomplished by Oracle Trading Community Model.

Editing Spoke Systems

The task of managing spoke systems involves the editing of the additional attributes that you associate with a spoke system. These attributes control data quality and import options.

Data Quality and Import Options

Data quality and import options defined for a spoke system are defaulted for a batch, thus eliminating the need to redefine them when you are creating batches.

Define Advanced Catalogs

What is catalog mapping?

You use Catalog Category mapping to map categories of different catalogs to the reporting categories in other catalogs. This feature allows one or more categories within a catalog to be mapped to category in a second catalog. For example, suppose that you want to roll up the costs associated with allow items assigned to a set of categories in catalog. Catalog mapping allows you to select a category in a catalog, and map all the categories in the set to that category. When you use this feature you are required to write code to do the roll up as identified in the example.

Define Data Quality for Products

Data Lens for Product Data Quality: Explained

The quality of product data is enhanced by the ability to check data quality while creating, editing, or importing items. This ability is provided by the integration of Oracle Fusion Product Hub with Oracle Product Data Quality. Oracle Product Data Quality provides classification, standardization, and matching of items, which are services that refine item data and prevent duplicate items.

In order to provide the ability to check data quality, you must initially define and later maintain a data lens in Oracle Product Data Quality, using metadata extracted from item classes and catalogs. A data lens uses semantic technology and contextual knowledge to clarify and cleanse product data as it enters a data repository.
Classification, Standardization, and Matching

Classification, standardization, and matching are the features that perform data quality checking.

In Oracle Product Data Quality, after you have initially set up the semantic model, you define rules for the following:

- **Classification**: rules that classify data items as belonging to the appropriate item class based on the values of certain attributes. Classification also includes suggested assignments of items to one or more catalogs and to categories within those catalogs.

- **Standardization**: rules that make the values of specified item attributes consistent with desired norms.

- **Matching**: rules for detecting when an item that is being created or imported matches an item that already exists in the Oracle Fusion Product Hub repository.

Extract Metadata for the Semantic Model

Before end users can check data quality in Oracle Fusion Product Hub, you must initially set up the semantic model in Oracle Product Data Quality. The semantic model is a fully configured data lens, which contains detailed contextual information about your data, and which contains your definitions for classification, standardization, and matching.

The foundation for the classification service is the hierarchy of your item classes and item catalogs. Consequently, you must set up the hierarchies for item classes and item catalogs, in Oracle Fusion Product Hub, before you extract metadata for building the semantic model.

Data quality depends primarily on the values of attributes. While defining an item class, select the attributes to be used when checking data quality, in the order that you want them to be applied for data quality checking. All selected attributes are used for classification and standardization. You can also select attributes to be used for matching duplicate items. In Oracle Product Data Quality, you later define the detailed rules for the relationships between these data quality attributes.

When the item hierarchies are in place, select the relevant catalogs and item classes on the Manage Catalogs or Manage Item Classes page, then select **Actions - Extract Data Quality Metadata**. This action schedules the Data Quality Metadata Extract process, which extracts metadata from the item class and catalog tables of Oracle Fusion Product Hub into an external file. The extract process includes the child hierarchy of the catalog or item class, and all of its attribute information. You can examine the results of the process on the Manage Scheduled Processes page, where you can also directly schedule the Data Quality Metadata Extract process.

In Oracle Product Data Quality, the AutoBuild feature uses extracted metadata for item classes and catalogs to construct the semantic model required for data quality checking.
You can also provide the AutoBuild feature with sample item data by exporting the results of an item search into an external spreadsheet, then transferring that spreadsheet data into Oracle Product Data Quality. To export the data, perform a search on the Manage Items page, then in the Search Results section, select **Actions - Export to Excel**.

For more information on transferring the extracted metadata, setting up the semantic model, and using AutoBuild, see the documentation for Oracle Product Data Quality, especially the following:

- Oracle Product Data Quality PIM Connector User's Guide
- Oracle Product Data Quality AutoBuild Reference Guide

**Maintaining the Data Lens**

As your data store evolves, you can continue to adjust the semantic model, modifying the rules, and extracting item class and catalog metadata.

**Profile Options**

The following profile options must be set when working with Oracle Product Data Quality.

<table>
<thead>
<tr>
<th>Display Name</th>
<th>Internal Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Product Data Quality DSA name</td>
<td>EGI_DSA_NAME</td>
<td>The name of the Oracle Product Data Quality DSA (Data Service Application) that is used to perform data quality checks on items. The default value is PRODUCT_HUB_DSA, which can be changed as needed.</td>
</tr>
<tr>
<td>Concurrent Requests for DSA</td>
<td>EGI_DSA_SERVER_THRESHOLD</td>
<td>The number of requests that the Oracle Product Data Quality DSA can support at a given time. This value is dependent on the server capacity on the user side.</td>
</tr>
</tbody>
</table>

**Data Security**

**Creating Data Privileges for Extensible Flexfield Attribute Groups: Explained**

To create data privileges for extensible flexfield attributes groups, log in to Oracle Authorization Policy Manager (APM) as IT Security Manager.

Item extensible flexfield attributes are supported by three data levels. Item, item revision, and item supplier, each with a translatable and non-translatable table as follows:
Non-Translatable Table

| Item Data Level                  | EGO_ITEM_EFF_B                |
| Item Revision Data Level         | EGO_ITEM_REVISION_EFF_B       |
| Item Supplier Data Level         | EGO_ITEM_SUPPLIER_EFF_B       |

Translatable Table

| Item Data Level Translatable     | EGO_ITEM_EFF_VL               |
| Item Revision Data Level Translatable | EGO_ITEM_REVISION_EFF_VL     |
| Item Supplier Data Level Translatable | EGO_ITEM_SUPPLIER_EFF_VL     |

Steps for Creating Data Privileges for Extensible Flexfield Attribute Groups in APM

- Step 1 - Open APM and in the Search region, enter the string: item and click the Search icon to find all database resources that begin with that item. For example, Item Data Level EGO_ITEM_extensible flexfield.

- Step 2 - From the results of the search, select the row for the database resource from table.

- Step 3 - Select Edit.

- Step 4 - On the Database Resource Details page, select the Action tab.

- Step 5 - Create data privileges for this table. For example, create two data privileges for Item Data Level Translatable table, one securing view-access to the attribute group and another securing edit-access (maintain) to the attribute group. Enter the name of the data grant of the attribute group you wish to secure. The name should not contain spaces. Enter the display name and description for the data grant. The display name will be shown in the user interface.

- Step 6 - Once the required privileges are created, submit the changes. A confirmation window indicates whether the update was successful.

Note

Data grants can be provisioned to individual users or groups of users at both item class and item levels.

Note

Data privileges can be created for view and edit operations. View item basic data, which provides user access to search for an item and view its basic item information has to be granted to the user for the user to access to the item extensible flexfield attribute groups.

Note

The same privilege can be used to secure multiple attribute groups or privileges can be created for each attribute group separately.
Associating Data Privileges with Extensible Flexfield Attribute Groups: Explained

After you’ve created data privileges in Oracle Authorization Policy Manager (APM), log into your Fusion Application to associate your data privileges.

Step 1. From the All Tasks tab in the Setup and Maintenance work area, in the Search region, search for Manage Item Attribute Groups task.

Step 2. In the Search Results table, highlight the row with Manage Item Attribute Groups and Attributes and select Go to Task.

Step 3. Search for attribute group that you want to secure.

Step 4. In the Search Results table, select the attribute group and click Edit.

Step 5. In the Context Usages section of the selected attribute group details page, access the View privilege column list of values and associate the view data privilege created in APM with this data level.

Note

Context values can be one of three values; Items, Item revision, or Item Supplier. Extensible flexfields can be created at all three.

Step 6. Access the Edit privilege column list of values and associate the Maintain data privilege created in APM with this data level.

Note

You can associate either read or edit privileges.

Step 7. Click Save and close the page.

Step 8. Go to Deploy Item Extensible Flexfields page to deploy the extensible flexfields.

Step 9. Launch the and enter EGO_ITEM_EFF into the Flexfield Code field in the Search region.

Step 10. Click Search.

Step 11. Select a row and click the Deploy Flexfield button to deploy the extensible flexfields.
Regional and Local Area Search

Regional Item Work Area Search

A regional search in the Item work area provides modes for general-item searches, including, Autosuggest, Key Word Search, and SQL searching.

The autosuggest option is enabled for the input field of the regional area.

**Regional Item Search Parameters**

The following are some of the parameters for Regional Item Search.

**Autosuggest Search**

Autosuggest is enabled for the item input text box in the Regional Area search. As you enter characters into the item input text box, suggested search topics are displayed in a drop-down list. Select a value from the drop-down list and click the Go icon to initiate your search.

**SQL Search**

SQL search is available only in the local area of the Item work area. To access it, click the Advanced link of the Item regional area search. For more details, see Local Area Item Search, below.

**Keyword Search**

Oracle Text technology provides keyword searching capabilities that uses data extracted from values in the items table and set up in an indexed file. The indexed data is set up by the administrator and is set up to run periodically. Additional indexes can be added by the administrator as needed. By default, regional search uses the following item attributes in the index:

- Item Number
- Description
- Item Class
- Long Description
• Organization Code
• Manufacturer

The Oracle Text search engine can be configured by the administrator to support features including:

• Word Stemming
• Wildcards
• Synonym Matches
• Fuzzy Searching
• Dictionary Matching
• Soundex Searching

Oracle Text is case insensitive.

Operators

The operators in the following table can be used with search expressions:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Operator</th>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operator Precedence</td>
<td>ABOUT</td>
<td>ACCUMulate (,)</td>
</tr>
<tr>
<td>AND (&amp;)</td>
<td>Broader Term (BT, BTG, BTP, BTI)</td>
<td>EQUIValence (=)</td>
</tr>
<tr>
<td>Fuzzy</td>
<td>HASPATH</td>
<td>INPATH</td>
</tr>
<tr>
<td>MINUS (-)</td>
<td>Narrower Term (NT, NTG, NTP, NTI)</td>
<td></td>
</tr>
<tr>
<td>NEAR (;)</td>
<td>NOT (~)</td>
<td>OR (</td>
</tr>
<tr>
<td>Related Term (RT)</td>
<td>soundex (!)</td>
<td></td>
</tr>
<tr>
<td>stem (s)</td>
<td>Stored Query Expression (SQE)</td>
<td>SYNonym (SYN)</td>
</tr>
<tr>
<td>threshold (&gt;)</td>
<td>Translation Term (TR)</td>
<td>Translation Term Synonym (TRSYN)</td>
</tr>
<tr>
<td>Top Term (TT)</td>
<td>weight (*)</td>
<td>wildcards (_%)</td>
</tr>
<tr>
<td>WITHIN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

You can use the following wildcards, which are features of Oracle Text search, to enhance your search:

Wildcards

• Percent sign (%) matches 0 or more characters
• Underscore (_) matches 0 or 1 character
• Backslash (\) escapes wildcard characters (\_% or \)
Word stemming allows word stems such as run to match other parts of speech such as ran, running, and runs. Word stemming requires that a dictionary be installed by the administrator and word stemming enabled for search. The operator stem ($), where $ is the word you wish to search using stemming, is used for stemming in search criterion.

**Fuzzy Operator**

The fuzzy operator is used to expand a query to include words that are spelled similarly to the term entered in the search criterion. The fuzzy operator is helpful for finding accurate results when there is frequent misspelling.

**Synonym Operator**

The synonym operator SYN enables the query to be expanded to include terms that are defined in a thesaurus as synonyms for specified terms.

**Soundex Operator**

The soundex operator ! enables the query to be expanded to include words that have similar sounds. This function enables comparison of words that are spelled differently, but sound alike in English or a 7-bit character set. For examples search '!Smythe' might return 'Smith'.

Oracle Text allows users to join keywords such as AND or OR within a single search criterion. Additionally, quotes can be used to indicate that characters within the quotes are to be searched in exact order.

**Within Operator**

The within operator allows the narrowing of a query down into document sections. Document sections can be one of the following:

- Zone Sections
- Field Sections
- Attribute Sections
- Special Sections (sentence or paragraph)

**ABOUT Operator**

The ABOUT operator allows the return of documents that are related to a query term or phrase. In English and French, ABOUT enables you to query on concepts, even if a concept is not actually part of a query. For example, an ABOUT query on heat might return documents related to temperature, even though the term temperature is not part of the query.

**AND and OR Operators**

The AND operator allows searching where at least one occurrence of each of the query terms exists. The OR operator allows searching where at least one occurrence of any of the query terms exist.

**Dictionary Matching**

Dictionary matching of the search terms can be enabled and requires a dictionary to be set up for the search. The spelling of terms in the input fields is checked against the dictionary to provide corrections to term misspellings.
Local Item Search Parameters

A local area search in the Item work area provides modes for general-item searches, including, Autosuggest, Key Word Search, and SQL searching.

**Local Area Item Search Parameters**

Item local area search is available in the Manage Items page and provides a search region in which users can enter values in one or more search fields, add search fields, and trigger the search.

The Manage Items search region has a default saved search, which defines the search fields that are displayed when the Manage Items page is initially opened. You can define your personal saved searches consisting of search fields to be used in the search. The seeded saved searches and personal saved searches are selected in the saved search drop list.

The Manage Item SQL follows the standards for search, but there are minor differences between the regional area or local area search and are listed below.

**Wildcards**

Multicharacter Wildcards: "%" and "*" must be interpreted as a wildcard that matches any character string, including spaces and special characters.

Single-Character Wildcards: "_" must be interpreted as a wildcard that matches any single character.

Literals: Literal "_", "%" and "*" characters must be preceded by the escape character "\"

---

**Note**

Wildcards as such "%" are not allowed at the start of a value and are not allowed for Item Number in the item search (due to performance reasons).

---

**Conditions for the Search Criteria**

As you select conditions for a search, drop-down lists will display only values that are appropriate to the scope of your search.

**Required Fields**

Some search fields are required. A blue asterisk indicates a require field. Two green asterisks indicate that at least one in a set of search fields is required.

**Keyword Search Field**

The Keyword search field is added to the search region to allow the user to search both with SQL and with keyword search. The keyword search field uses the same features as the regional area search, except that the autosuggest feature is not available. When a user enters values for both the keyword search fields and other search fields, the keyword search and SQL search are performed separately. The results from both are displayed in the search result table.
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.

<table>
<thead>
<tr>
<th>Product</th>
<th>Job or Abstract Role</th>
<th>Security Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Fusion Assets</td>
<td>Asset Accountant</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Assets</td>
<td>Asset Accounting Manager</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Enterprise Planning and Budgeting</td>
<td>Budget Analyst</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Enterprise Planning and Budgeting</td>
<td>Budget Manager</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Financial Consolidation Hub</td>
<td>Consolidation Accountant</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Financial Consolidation Hub</td>
<td>Consolidation Manager</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Financials Common Module</td>
<td>Intercompany Accountant</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion General Ledger</td>
<td>Financial Analyst</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion General Ledger</td>
<td>General Accountant</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion General Ledger</td>
<td>General Accounting Manager</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Incentive Compensation</td>
<td>Incentive Compensation</td>
<td>View Manager Hierarchy</td>
</tr>
<tr>
<td></td>
<td>Participant Manager</td>
<td></td>
</tr>
<tr>
<td>Oracle Fusion Inventory Management</td>
<td>Warehouse Manager</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Project Foundation</td>
<td>Project Accountant</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Project Foundation</td>
<td>Project Administrator</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Project Foundation</td>
<td>Project Billing Specialist</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Purchasing</td>
<td>Buyer</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Sourcing</td>
<td>Category Manager</td>
<td>View All Workers</td>
</tr>
<tr>
<td>Oracle Fusion Sourcing</td>
<td>Sourcing Project Collaborator</td>
<td>View All Workers</td>
</tr>
</tbody>
</table>

For example, as part of their Transactional Business Intelligence setup:

- Oracle Fusion Assets implementors must assign the predefined security profile View All Workers to the Asset Accountant and Asset Accounting Manager job roles.

- Oracle Fusion Incentive Compensation implementors must assign the predefined security profile View Manager Hierarchy to the abstract role Incentive Compensation Participant Manager.

The security profiles that HCM roles need to access Transactional Business Intelligence are assigned during the setup of HCM data security: no additional setup is required for Transactional Business Intelligence purposes.
Enabling an Oracle Fusion Transactional Business Intelligence User to Access Person Data: Worked Example

This example shows how to assign a security profile to a job or abstract role to enable users with that role to access person data. This task is required for users of Oracle Fusion Transactional Business Intelligence (Transactional Business Intelligence) who do not also use Oracle Fusion Human Capital Management (HCM).

The following table summarizes key decisions for this scenario. When performing this task, use the job or abstract role for your product and the name of the relevant predefined person security profile in place of those shown here.

<table>
<thead>
<tr>
<th>Decisions to Consider</th>
<th>In This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the name of the Transactional Business Intelligence job or abstract role?</td>
<td>Warehouse Manager</td>
</tr>
<tr>
<td>What is the name of the person security profile?</td>
<td>View All Workers</td>
</tr>
</tbody>
</table>

Summary of the Tasks

To perform these tasks, you must have the role IT Security Manager.

1. Launch the task Manage Data Role and Security Profiles.
2. Search for the job or abstract role.
3. Assign the relevant predefined security profile to the job or abstract role.

Launching the Task Manage Data Role and Security Profiles

1. On the Overview page of the Setup and Maintenance work area, click the All Tasks tab.
2. In the Search region, complete the fields as shown in this table:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search</td>
<td>Tasks</td>
</tr>
<tr>
<td>Name</td>
<td>Manage Data Role and Security Profiles</td>
</tr>
</tbody>
</table>

3. Click Search.
4. In the search results, click Go to Task for the Manage Data Role and Security Profiles task.

Searching for the Job or Abstract Role

1. On the Manage HCM Data Roles page, enter the job name Warehouse Manager in the Role field.
2. Click Search.
3. In the search results, highlight the entry for the Warehouse Manager job role.

4. Click Assign.

Assigning the Security Profile to the Job Role

1. In the Person Security Profile field on the Assign Data Role: Security Criteria page, select the security profile View All Workers.

2. Click Review.

3. On the Assign Data Role: Review page, click Submit.

Define Extensions: 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 described in the Oracle Fusion Applications Administrator’s Guide and Oracle Fusion Applications Extensibility Guide for Developers. As you read content from these guides, note that the guides mention 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.

Note

Oracle Enterprise Manager Fusion Applications Control is not available for Oracle Cloud implementations.

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, Creating, and Editing Job Definitions**

• You can access predefined and custom job definitions. In the table on the Manage Job Definitions tab, the Name column displays an asterisk for predefined job definitions. Refer to the Oracle Fusion Applications Administrator's Guide.

See: Viewing Job Definitions

• You or a technical administrator can create jobs based on Java, PL/SQL, Oracle Business Intelligence (BI) Publisher, or any other supported technology. Every predefined or custom job must have a job definition. For Oracle Cloud implementations, custom job definitions can be created only for custom jobs based on Oracle BI Publisher reports. Refer to the Oracle Fusion Applications Administrator's Guide.

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.

• You can edit all aspects of custom job definitions. For predefined job definitions, you can’t update parameters, but you can determine if user properties are read-only or not. You can also edit certain aspects of
predefined definitions, which are described as job properties in the Oracle Fusion Applications Extensibility Guide for Developers.

See: Customizing Existing Oracle Enterprise Scheduler Job Properties

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 application defined properties of job definitions. For example, you can use a source of country values for a Country job parameter.

---

Note

Since parameters for predefined job definitions cannot be edited, list of values sources are only for parameters in custom job definitions.

---

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

---

Contextual Addresses

Setting Up the Mapping Service for Contextual Addresses: Points to Consider

A contextual address is marked with an orange square contextual action icon that can be clicked to display the address on a map. Use the Mapping Service for Contextual Addresses profile option to specify the web mapping service to be used to display the map. In the Setup and Maintenance work area, go to the Manage Administrator Profile Values task to set the profile option value.
Profile Option Default

By default, the Mapping Service for Contextual Addresses profile option has no value. Until you enter a valid value for this profile option, users get an error when they try to open a map for any contextual address.

Profile Option Value

After you find and select the Use the Mapping Service for Contextual Addresses profile option, enter a mapping service URL in the Profile Value column, for example:

- http://maps.yahoo.com/maps_result.php?q1=
- http://bing.com/maps/?v=2&encType=1&where1=

Optionally, add parameters to the URL. For example, to avoid a locator box in Google Maps, add &iwloc=& to the URL, so that you enter http://maps.google.com/maps?iwloc=&amp;&output=embed&q= as the profile value.

FAQ for Privacy Statement

How can I enable the privacy statement?

Use the Privacy Statement URL profile option to enable the Privacy Statement menu item in the global area. This menu item in the Settings and Actions menu is disabled by default.

Open the Setup and Maintenance work area, and use the Manage Applications Core Administrator Profile Values task to find the Privacy Statement URL profile option. In the Profile Value column, enter the full URL of the web page that contains the privacy content you want the menu item to link to.
Web Services: Overview

Use web services to integrate web-based applications into your Oracle Fusion applications. Web services expose Oracle Fusion Applications business objects and processes to other applications through the use open standards-based technologies. Some of these technologies include Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Business Process Execution Language (BPEL), Web Services Description Language (WSDL), and XML schema definitions (XSD). Oracle Fusion Applications web services support development environments and clients that comply with these open standards.

Oracle Fusion Applications includes two types of web services: Application Development Framework (ADF) services and composite services. The following table describes the two types.

<table>
<thead>
<tr>
<th>Web Service Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADF services</td>
<td>ADF services usually represent business objects, such as employees or purchase orders. ADF services typically expose standard operations, such as create, update, and delete. However, for locally-persisted objects, ADF services are not limited to these operations. Examples of ADF services include:</td>
</tr>
<tr>
<td></td>
<td>• Worker.changeHireDate - a service that updates the hire date of the worker business object.</td>
</tr>
<tr>
<td></td>
<td>• ProjectTask.createTask - a service that adds a task to the project task business object.</td>
</tr>
<tr>
<td>Composite services</td>
<td>Composite services usually represent end-to-end business process flows that act on business events produced by the ADF services. Composite services orchestrate multiple object-based services, rules services, and human workflows. Examples of composite services include:</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>-  ProjectStatusChangeApproval.process - a service that accepts the change in project status.</td>
</tr>
<tr>
<td></td>
<td>- ScheduleOrchestrationOrderFulfillmentLineService.scheduleOrders - a service that schedules resources used to fulfill an order.</td>
</tr>
</tbody>
</table>

Access Oracle Enterprise Repository for Oracle Fusion Applications to find detailed information about integration assets, such as web services. To view lists of web services, select these asset types:

- ADF Service
- ADF Service Data Object
- Composite Service
- Composite

Service methods and parameters, the service path, the WSDL URL and other technical data, appear on the Detail tab of each web service. Step-by-step instructions regarding the invocation of a service and the service XSD appear on the Documentation tab.

**Files for Import and Export**

**Files for Import and Export: Explained**

You can import data into or export data out of Oracle Fusion Applications using repositories of content and processes for import and export.

Integration specialists stage data for import and export. Application administrators run processes to import data in repositories of content to application transaction tables, or retrieve data exported from applications.

Aspects of managing files for import and export involve the following:

- The File Import and Export page
- Interacting with content management
- Uploading for import
- Downloading for export
The File Import and Export Page

The File Import and Export page lets you upload content to or download content from the document repository of Oracle WebCenter Content Management. For information or assistance regarding general access to content management (including all metadata), to create and manage accounts, and to programmatically upload and download content, contact the WebCenter Content Administrator.

Search criteria on the page are limited to the minimum metadata of content management records needed for file import and export.


Interacting with Content Management

Everyone who uses the File Import and Export page is assigned to one or more accounts in content management.

Accounts organize and secure access to content items.

Uploading for Import

Uploading a file creates a record.

When you create a record, you must specify an account as well as the file. When you create a record, you must specify an account as well as the file. The account you specify determines which import process picks up that file to import it.

You can upload any file formats that can be parsed by the content repository being used, such as any MIME or content types. However, the format uploaded should conform to the requirements of the import process being used, such as a comma-separated values (CSV) file for the Load Interface File for Import process.

Downloading for Export

Processes you run to export data result in files in content management. Records in the search results table of the File Import and Export page provide links to the files for download.

Note

The owner of a data export file can be an application ID (APPID).

File Size

Upload and download does not intentionally apply the following:

- Data compression
- File chunking or splitting
The **UPLOAD_MAX_DISK_SPACE** parameter in the **web.xml** file determines the maximum allowable file size in content management. The default maximum size is 10240000 (10MB).

**Files for Import and Export: Points to Consider**

Interaction between the File Import and Export page and Oracle WebCenter Content requires securing content in an account. Oracle provides predefined accounts in Oracle WebCenter Content.

Areas of file import and export to consider involve the following:

- Security
- Searching records
- Accessing content in a new account
- Account names
- Deleting files

**Security**

The duty role needed for accessing the File Import and Export page is File Import and Export Management duty. This duty role is included in the predefined role hierarchy for integration specialist roles and product family administrator roles.

Files in Oracle WebCenter Content are associated with an account so that only users who have permission to a particular account can work with content items that belong to that account. You can only upload and download files to and from content management that are associated with accounts that you are entitled to access.

Oracle WebCenter Content does not support trailing slashes (/). Account names are appended with a $ to ensure each account is unique. Account names are dynamic so that if they overlap (one name is completely contained in another, longer name, such as US and USSales), each account is treated as discrete by access grants.

Security such as virus scanning is handled by the underlying integrated content management.

**Searching Records**

A record in Oracle WebCenter Content contains metadata used for accessing the file.

When a scheduled process has run to completion on a file, the record for the file includes a process ID.

**Accessing Content in a New Account**

When you create a new account in Oracle WebCenter Content and the content server is not restarted, access to content in the new account from the File Import and Export page may be delayed until the policy store is updated.
**Account Names**

If you create custom accounts for importing or exporting data, use the following conventions for naming the account: Do not include a slash "/" at the beginning or end. End with "$" to avoid partial string matching. Use "/$" as a separator in the hierarchical structure.

For example: `fin$/journal$/import$`. The File Import and Export page transforms account names by removing the $s. For example, `fin$/journal$/import$` displays as `fin/journal/import`. The Remote Introduc Client (RIDC) HTTP command-line interface (CLI) transforms the account name you specify without $ symbols to one that includes them. For example, `fin/journal/import` becomes `fin$/journal$$/import$` in WebCenter Content.

**Deleting Files**

You can delete one file at a time when you use the File Import and Export page. To delete multiple files simultaneously from the content repository, use the standard service page in Oracle WebCenter Content.

**External Data Integration Services for Oracle Cloud**

**External Data Integration Services for Oracle Cloud: Overview**

Use External Data Integration Services for Oracle Cloud to load data into Oracle Fusion Applications from external sources, such as legacy systems and third-party applications.

Components of External Data Integration Services for Oracle Cloud include:

- Templates and control files for formatting, structuring, and generating the data file.
- A general file load process for loading values from the data file into interface tables.
- Application-specific data import processes for transferring data from interface tables to the application tables in your Oracle Fusion Applications.

To use External Data Integration Services for Oracle Cloud to load data into Oracle Fusion Applications tables:

1. Prepare your data and generate a data file by using the product-specific templates and control files.
2. Transfer the data file to the integrated content management server.
3. Run the Load Interface File for Import process.
4. Correct data load errors, if necessary.
5. Run the appropriate application-specific process for validating and inserting the data into application tables.

6. Correct data import errors, if necessary.

For templates and control files, see assets with the File-Based Data Import type in Oracle Enterprise Repository for Oracle Fusion Applications (http://fusionappsoer.oracle.com). For more information, see the Documentation tab for the Load Interface File for Import process in Oracle Enterprise Repository.

### Locating File Import Templates: Explained

External data that you integrate into your Oracle Fusion Applications must be structured and formatted according to the properties of the fields and tables that store the data. To prepare external data so that data types, structural relationships, and other properties of the data correctly align to the data types, structural relationships, and properties of the target tables, use the product-specific templates and control files in Oracle Enterprise Repository for Oracle Fusion Applications.

You access these files from the Documentation tab of the scheduled process that corresponds to the interface tables that store the data. To find the process, you can search the interface table or you can search the specific process, if you know it.

Aspects of preparing external data using templates involve these tasks.

- Finding templates and control files
- Downloading templates
- Opening XLS templates

### Finding Templates and Control Files

To find the templates and control files:

1. Sign in to Oracle Enterprise Repository.
2. Enter the following information in the Search fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search String</td>
<td>FBDI</td>
</tr>
<tr>
<td>Type</td>
<td>Scheduled Process</td>
</tr>
<tr>
<td>FusionApps: Logical Business Area</td>
<td>(Optional) Select the value relevant to your implementation.</td>
</tr>
</tbody>
</table>

3. Click Search.
4. Select Load Interface File for Import from the results.

### Downloading Templates

To download the templates:
1. Use the Search area to locate the Load Interface File for Import job and then select it from the search results.

2. Click the Documentation tab in the lower pane to see a list of links to application-specific import jobs.

3. Click a link to access the job.

4. Click the Documentation tab in the lower pane to see a list of links that access:
   
   - Control files, which describe the logical flow of the data load process
   - XLS templates, which include worksheets and macros that assist you in structuring, formatting, and generating your data file

5. Click the link to download the file.

Opening the XLS Template

To prepare your data in a spreadsheet format, use XLS templates:

1. Open the XLS template.
   
The first worksheet in each file provides instructions for using the template.

**Important**

If you omit or fail to complete the instructions, data load errors and data import failure are likely.

2. Save a copy of the file.

3. Click the **Generate CSV File** button.
   
The macro generates a comma-separated values (CSV) file and compresses it into a ZIP file; you must transfer the ZIP file to the content management server.

Using Excel Integration Templates to Generate Data Files: Points to Consider

Oracle Enterprise Repository for Oracle Fusion Applications includes integration templates to help you prepare external data for loading and importing. Each template includes table-specific instructions, guidelines, formatted spreadsheets, and best practices for preparing the data file for upload. Use the templates to ensure that your data conforms to the structure and format of the target application tables.

**Templates**

This list details the characteristics of the templates:

- Each interface table is represented by a separate worksheet.
• Each interface table field is represented by a worksheet column with a header in the first row.

• Each column header contains bubble text, or comments, that include details about the column, such as the expected data type, length, and, in some cases, other instructional text.

• The worksheet columns appear in the order that the control file processes the data file.

• The columns that you do not intend to use can be hidden, but not reordered or deleted.

**Important**
Deleting or reordering columns will cause the load process to fail and result in an unsuccessful data load.

• The external data must conform to the data type that the control file and process for the associated database column accepts.

• Date column values must appear in the YYYY/MM/DD format.

• Amount column values must appear with no separators other than a period (.) as the decimal separator.

• Negative values must be preceded by the minus (-) sign.

• Column values that require whole numbers include data validation to allow whole numbers only.

• Columns are formatted, where applicable, to match the target field data type to eliminate data entry errors.

• For columns that require internal ID values, refer to the bubble text for additional guidance about finding these values.

• When using Microsoft Excel to generate or update the CSV file, you must select YYYY/MM/DD as your regional setting for date values.

**Using XML Templates to Generate Data Files for Integration:**

**Highlights**

Oracle Enterprise Repository for Oracle Fusion Applications includes XML integration templates assets that you use with Oracle Data Integrator (ODI) to generate import files from your external data.

To use the XML templates and generate the import files, you must:

• Install and set up Oracle Data Integrator

• Create source and target models

• Create integration projects

**Note**
In Oracle Cloud implementations, you must upload the ZIP file to the content management repository in Oracle Cloud. In non-Cloud implementations, you can streamline the data integration process by installing the content management document transfer utility so ODI performs the ZIP file transfer.

Oracle Data Integrator provides a solution for integrating complex data from a variety of sources into your Oracle Fusion applications. The Oracle Fusion Middleware Installation Guide for Oracle Data Integrator and the Oracle Fusion Middleware Developer’s Guide for Oracle Data Integrator provide complete details pertaining to the installation and set up of this product.

**Installing and Setting Up Oracle Data Integrator**

- Install Oracle Data Integrator to use Oracle Fusion Applications XML integration templates. Refer to the Oracle Fusion Middleware Installation Guide for Oracle Data Integrator.
  
  See: Installing Oracle Data Integrator

- Set up Oracle Data Integrator to use Oracle Fusion Applications XML integration templates. Refer to the Oracle Fusion Middleware Developer’s Guide for Oracle Data Integrator.
  
  See: Setting up the Topology

**Creating Source and Target Models**

- Create the ODI models for both the source and target datastores. You determine the source models that you use based on the system or technology of the external data that you import into your Oracle Fusion application. You create the target models by importing the XML files, which you download from Oracle Enterprise Repository. For more information, refer to the Oracle Fusion Middleware Developer’s Guide for Oracle Data Integrator.
  
  See: Creating and Reverse-Engineering a Model

**Configuring Integration Projects**

- Create and configure an integration project, which entails selecting the knowledge modules, creating the interfaces, and mapping the source and target datastores. For more information, refer to the Oracle Fusion Middleware Developer’s Guide for Oracle Data Integrator.
  
  See: Creating an Integration Project

**Creating Integration Projects That Generate Data Files for Import: Points to Consider**

When you use Oracle Data Integrator (ODI) to generate import data files from external data sources, you must configure an integration project. Integration projects are collections of ODI components that provide the procedural details of an integration from a source to a target. In this scenario, the source is your
external data and the target is the import data file that you load and import into your Oracle Fusion Applications.

Configure these components to create your integration project.

- Knowledge modules
- Integration interfaces

**Knowledge Modules**

Knowledge modules contain the information that Oracle Data Integrator requires to perform a specific set of tasks against a specific technology or set of technologies. Examples include Check knowledge modules, which ensure that constraints on the sources and targets are not violated and integration knowledge modules, which load data to target tables.

Consider the following points about knowledge modules:

- The knowledge modules that you import into your integration project depend on the source and target technologies, as well as other integration-specific rules and processes.
- Multiple types of knowledge modules exist in ODI.
- Use the SQL File to Append module to create the import data file.

**Integration Interfaces**

Integration interfaces comprise sets of rules that define the loading of data from one or more sources to the target.

Consider the following points about integration interfaces:

- The source is the datastore from your external data model.
- The target is the interface table datastore, which is the CSV file from your interface table model.
- After you set up the source and target datastores, map the target fields to the source fields.
- You can map source field values to target fields or constants.

**Transferring Data Files to Target Accounts in Oracle WebCenter Content: Explained**

After you generate the ZIP file that contains the CSV data import file, transfer it to the content repository.

Use any of these methods to transfer file:

- File Import and Export page in Oracle Fusion Applications
- Oracle WebCenter Content Document Transfer Utility
- Oracle Fusion Financials Utility web service
Note
Consult Oracle Enterprise Repository for Oracle Fusion Applications for web service documentation.

Aspects of transferring data files to content management involve the following:

- Target accounts
- Accessing transferred content

**Target Accounts**

You must transfer files to these predefined account in content management that corresponds to the interface table or assets.

<table>
<thead>
<tr>
<th>Interface Table</th>
<th>Predefined Account</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payables Standard Invoice Import</td>
<td>fin/payables/import</td>
</tr>
<tr>
<td>• AutoInvoice Import</td>
<td>fin/receivables/import</td>
</tr>
<tr>
<td>• Receivables Standard Receipt Import</td>
<td>fin/receivables/import</td>
</tr>
<tr>
<td>• Customer Import</td>
<td>fin/receivables/import</td>
</tr>
<tr>
<td>• China Value Added Tax Invoice Import</td>
<td>fin/receivables/import</td>
</tr>
<tr>
<td>• BAI2 Format Bank Statements Import</td>
<td>fin/cashManagement/import</td>
</tr>
<tr>
<td>• EDIFACT FINSTA Format Bank Statements Import</td>
<td>fin/cashManagement/import</td>
</tr>
<tr>
<td>• ISO200022 CAMT053 Format Bank Statements Import</td>
<td>fin/cashManagement/import</td>
</tr>
<tr>
<td>• SWIFT MT940 Format Bank Statements Import</td>
<td>fin/cashManagement/import</td>
</tr>
<tr>
<td>• Fixed Asset Mass Additions Import</td>
<td>fin/assets/import</td>
</tr>
<tr>
<td>• Fixed Asset Mass Adjustments Import</td>
<td>fin/assets/import</td>
</tr>
<tr>
<td>• Fixed Asset Mass Retirements Import</td>
<td>fin/assets/import</td>
</tr>
<tr>
<td>• Fixed Asset Mass Transfers Import</td>
<td>fin/assets/import</td>
</tr>
<tr>
<td>• Fixed Asset Units of Production Import</td>
<td>fin/assets/import</td>
</tr>
<tr>
<td>Intercompany Transaction Import</td>
<td>fin/intercompany/import</td>
</tr>
<tr>
<td>• Journal Import</td>
<td>fin/generalLedger/import</td>
</tr>
<tr>
<td>• Chart Of Accounts Segment Values and Hierarchies Import</td>
<td>fin/generalLedger/import</td>
</tr>
<tr>
<td>General Ledger Budget Balance Import</td>
<td>fin/budgetBalance/import</td>
</tr>
<tr>
<td>Supplier Bank Account Import</td>
<td>fin/payables/import</td>
</tr>
<tr>
<td>Tax Configuration Content Import</td>
<td>fin/tax/import</td>
</tr>
<tr>
<td>Import Blanket Purchase Agreements</td>
<td>prc/blanketPurchaseAgreement/import</td>
</tr>
<tr>
<td>Import Contract Purchase Agreements</td>
<td>prc/contractPurchaseAgreement/import</td>
</tr>
<tr>
<td>Import Purchase Orders</td>
<td>prc/purchaseOrder/import</td>
</tr>
<tr>
<td>Import Requisitions</td>
<td>prc/requisition/import</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>• Import Suppliers</td>
<td>prc/supplier/import</td>
</tr>
<tr>
<td>• Import Supplier Sites</td>
<td></td>
</tr>
<tr>
<td>• Import Supplier Site Contacts</td>
<td></td>
</tr>
<tr>
<td>• Import Supplier Site Assignments</td>
<td></td>
</tr>
<tr>
<td>Project Enterprise Resource Import</td>
<td>prj/projectManagement/import</td>
</tr>
<tr>
<td>Project Unprocessed Expenditure Item Import</td>
<td>prj/projectCosting/import</td>
</tr>
<tr>
<td>Cycle Count Import</td>
<td>scm/cycleCount/import</td>
</tr>
<tr>
<td>Inventory Reservation Import</td>
<td>scm/inventoryReservation/import</td>
</tr>
<tr>
<td>Inventory Transaction Import</td>
<td>scm/inventoryTransaction/import</td>
</tr>
<tr>
<td>Item Import</td>
<td>scm/item/import</td>
</tr>
<tr>
<td>Receiving Receipt Import</td>
<td>scm/receivingReceipt/import</td>
</tr>
<tr>
<td>Shipment Request Import</td>
<td>scm/shipmentRequest/import</td>
</tr>
</tbody>
</table>

You can create subaccounts to further organize your files. However, you must create the account subordinate to the predefined account for the asset you are integrating.

### Accessing Transferred Content

To access your transferred data you must access the account that corresponds to the interface table or asset appropriate for the data.

Available data integration processes move the content into and out of Oracle Fusion Applications tables. Running an import or export process creates a process ID in content management that you can use to identify the content you wish to overwrite or extract.

Oracle Enterprise Scheduler import process jobs result in the following hierarchy of items in Oracle WebCenter Content:

- A root import job is a list of all unprocessed files in an account. This job submits the child jobs that process each unprocessed file.
- A parent import job is a single file ID, account name, and the import steps (download, extract, import) for a single job, job set, or subrequests. This type of job tags the file with its request ID, provided the file is not deleted immediately after successful import.
- A child import job is a direct data load from a prepared file, typically a SQLLoader. Typically, the parent import job submits this job.

### Document Transfer Utility: Explained

The WebCenter Content Document Transfer Utility for Oracle Fusion Applications is a feature-set Java library that provides programmatic access to
the content repository. Use the utility to import and export documents, such as import files that contain external data that you want to load into interface and application tables.

The library includes:

- Oracle WebCenter Content client command line tool
- Oracle Data Integrator (ODI) upload and download tools
- Oracle WebCenter Content remote intradoc client (RIDC)
- Oracle HTTPClient
- Oracle Fusion Applications branding and defaults

Options for the WebCenter Content Document Transfer Utility for Oracle Fusion Applications fall into these categories:

- DownloadTool program options
- UploadTool program options
- Debugging and silent invocation options

**DownloadTool Program Options**

This table describes the download tool program options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>Protocol-specific connection URL of content server</td>
</tr>
<tr>
<td>username</td>
<td>Username to leverage</td>
</tr>
<tr>
<td>password</td>
<td>Password, supplied in command line</td>
</tr>
<tr>
<td>passwordFile</td>
<td>Password, supplied in text file on the first line of the file</td>
</tr>
<tr>
<td>dID</td>
<td>ID of document revision to download</td>
</tr>
<tr>
<td></td>
<td>dID is unique across repository</td>
</tr>
<tr>
<td></td>
<td>dID changes with each revision</td>
</tr>
</tbody>
</table>

**Note**

Alternatively, specify the dDocName and RevisionSelectionMethod to identify the dID to leverage.

<table>
<thead>
<tr>
<th>Option</th>
<th>Content name</th>
</tr>
</thead>
<tbody>
<tr>
<td>dDocName</td>
<td>Multiple revisions of a document can share the same dDocName value, otherwise it is unique.</td>
</tr>
</tbody>
</table>

**Note**

You should also provide RevisionSelectionMethod value.
RevisionSelectionMethod | Revision to download  
|-------------------------|------------------  
| Valid values: Latest, LatestReleased  
| Default value: Latest  
outputFile | Path and name of local file to write  

Here you see a sample download invocation command:

```java
java -classpath "oracle.ucm.fa_client_11.1.1.jar"
oracle.ucm.client.DownloadTool
url=http://ucmserver.com:16200/cs/idcplg username=weblogic
password=welcome3i
dID=21537 outputFile="/tmp/output.doc"
```

Here you see sample output:

Oracle WebCenter Content Document Transfer Utility
Oracle Fusion Applications
Copyright (c) 2013, Oracle. All rights reserved.
Performing download (GET_FILE) ...  
Download successful.

**UploadTool Program Options**

This table describes the upload tool program options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>Protocol-specific connection URL of content server</td>
</tr>
<tr>
<td>username</td>
<td>Username to leverage</td>
</tr>
<tr>
<td>password</td>
<td>Password, supplied in command-line</td>
</tr>
<tr>
<td>passwordFile</td>
<td>Password, supplied in text file on the first line of the file</td>
</tr>
<tr>
<td>primaryFile</td>
<td>Fully-qualified path of local primary file to upload</td>
</tr>
<tr>
<td>dDocAccount</td>
<td>Destination account</td>
</tr>
<tr>
<td>dDocTitle</td>
<td>Document title</td>
</tr>
<tr>
<td>checkout</td>
<td>If uploading a document revision, check out the document from the repository before uploading the revision</td>
</tr>
<tr>
<td></td>
<td>Valid values: true, false</td>
</tr>
<tr>
<td></td>
<td>Default value: false</td>
</tr>
<tr>
<td>ignoreCheckoutErrorNeg22</td>
<td>Ignore error -22 (user has already checked-out the document) when checking-out the document.</td>
</tr>
<tr>
<td></td>
<td>Valid values: true, false</td>
</tr>
<tr>
<td></td>
<td>Default value: true</td>
</tr>
</tbody>
</table>

Here you see a sample upload invocation command:

```java
java -classpath "oracle.ucm.fa_client_11.1.1.jar"
oracle.ucm.client.UploadTool
url=http://ucmserver.com:16200/cs/idcplg username=weblogic
password=welcome3i
```
Here you see sample output:

Oracle WebCenter Content Document Transfer Utility
Oracle Fusion Applications
Copyright (c) 2013, Oracle. All rights reserved.
Performing upload (CHECKIN_UNIVERSAL) ...
Upload successful.
[dID=21537 | dDocName=UCMFA021487]

Debugging and Silent Invocation Options

This table describes the options which are common to all tools that the invoker can leverage.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>verbose</td>
<td>Verbose output</td>
</tr>
<tr>
<td>quiet</td>
<td>Minimal output</td>
</tr>
<tr>
<td>version</td>
<td>Print tool revision or version</td>
</tr>
<tr>
<td>log_file_name</td>
<td>Send program output to specified log file instead of the System.out log file</td>
</tr>
<tr>
<td>log_file_append</td>
<td>Append log to existing log file rather than overwrite it</td>
</tr>
<tr>
<td>Valid values:</td>
<td>true, false</td>
</tr>
<tr>
<td>Default value:</td>
<td>false</td>
</tr>
<tr>
<td>socketTimeout</td>
<td>Override time out of socket</td>
</tr>
<tr>
<td>Specify override time in seconds</td>
<td></td>
</tr>
</tbody>
</table>

You can use the tools to test the connection. Provide only the url, username, and password as you see in this sample test:

```java
java -classpath "oracle.ucm.fa_client_11.1.1.jar"
oracle.ucm.client.DownloadTool
url=http://ucmserver.com:16200/cs/idcplg username=weblogic
password=we1com3l
```

Here you see the sample output:

Oracle WebCenter Content Document Transfer Utility
Oracle Fusion Applications
Copyright (c) 2013, Oracle. All rights reserved.
Performing connection test (PING_SERVER) ...
Connection test successful.

Load Interface File for Import Process

Use to load external setup or transaction data from a data file in the content repository to interface tables. The process prepares the data for import into application tables.
You run this process from the Scheduled Processes page. You can run it on a recurring basis.

Before running this process, you must:

1. Prepare your data file.
2. Transfer the data file to the content repository.

**Parameters**

**Import Process**

Select the target import process.

**Data file**

Enter the relative path and the file name of the *.zip data file in the content repository.

**Importing Data into Application Tables: Procedure**

The final destination for your external data is the application data tables of your Oracle Fusion Applications product.

Aspects of importing data into application tables involve the following:

- Loading data into interface tables
- Finding and submitting the import process

**Loading Data into Interface Tables**

Interface tables are intermediary tables that store your data temporarily while the system validates format and structure. Run the Load Interface File for Import scheduled process to load data from the data file into the interface table that corresponds to the template that you use to prepare the data.

To load your data into interface tables, submit the Load Interface File for Import scheduled process:

1. Sign in to Oracle Fusion Applications.
2. In the Navigator menu, select Tools, Scheduled Processes
3. Click the Schedule New Process button.
4. Search and select the Load Interface File for Import job.
5. When the Process Details page appears:
   a. Select the target import process.
   b. Enter the data file name.

**Note**
If the file exists in an account subordinate to the predefined account, you must enter the entire path relative to the predefined account in the content repository. Include all subaccounts and the file name.

6. Submit the process.

If no errors exist in the data file, then the process populates the interface tables.

**Note**
The data file remains in the content repository after the process ends.

### Finding and Submitting the Import Process

Run the appropriate import process to import the data into the interface tables of your Oracle Fusion Applications product.

**To import your data:**

1. Sign in to Oracle Fusion Applications.
2. In the Navigator menu, select Tools, Scheduled Processes
3. Click the Schedule New Process button.
4. Find and select the import process that is specific to the target application tables.
5. When the Process Details page appears, select the process that corresponds to the data that you are importing.

   If you prepared your data using the spreadsheet template, select the process named in the Overview section of the spreadsheet.

6. Submit the process.

**Note**
For more detailed information on the process used for data prepared using the spreadsheet template, see the Instructions and CSV Generation tab of the spreadsheet template.

### Correcting Import Load Process Errors: Explained

The Load Interface File for Import process ends in error if the load of the data file fails on any row.

The following conditions apply when the process ends in error:

- The Load File to Interface child process ends in either warning or error.
• All rows that were loaded by the process are deleted, even those rows that loaded successfully.

To correct errors:

1. Review the error logs.
2. Change any formatting or structural anomalies that exist in the data.
3. Recreate the CSV and ZIP files.
4. Transfer the file to the content management server.
5. Submit the Load Interface File for Import job.
6. Repeat these steps until the process successfully loads the data.
7. Import the data using the appropriate product-specific process.
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 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.
- The Oracle Social Network business objects and the Navigator menu customizations are moved using the customization sets on the Customization Migration page instead of using the export and import function in the Setup and Maintenance work area.

<table>
<thead>
<tr>
<th>Business Object Name</th>
<th>Moved Functional Item</th>
<th>Effect on the Scope of Movement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Message</td>
<td>Messages and associated tokens</td>
<td>No parameters: all messages are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>moduleType/moduleKey:</strong> only messages belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>messageName/applicationId:</strong> only the specified message is moved.</td>
</tr>
<tr>
<td>Application Taxonomy</td>
<td>Application taxonomy modules and components</td>
<td>No parameters: all taxonomy modules and components are moved.</td>
</tr>
<tr>
<td>Application Attachment Entity</td>
<td>Attachment entities</td>
<td>No parameters: all attachment entities are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>moduleType/moduleKey:</strong> only attachment entities belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</td>
</tr>
<tr>
<td>Application Attachment Category</td>
<td>Attachment categories and category-to-entity mappings</td>
<td>No parameters: all attachment categories and category-to-entity mappings are moved. 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.</td>
</tr>
<tr>
<td>Application Document Sequence Category</td>
<td>Document sequence categories</td>
<td>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.</td>
</tr>
<tr>
<td>Application Document Sequence</td>
<td>Document sequences and their assignments</td>
<td>No parameters: all sequences are moved. moduleType/moduleKey: only document sequences belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved name: only the specified document sequence is moved.</td>
</tr>
<tr>
<td>Application Descriptive Flexfield</td>
<td>Descriptive flexfield registration data and setup data</td>
<td>No parameters: all descriptive flexfields are moved. moduleType/moduleKey: only descriptive flexfields belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved. descriptiveFlexfieldCode/applicationKeyId: only the specified descriptive flexfield is moved.</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Only flexfields with a deployment status of Deployed or Deployed to Sandbox are eligible to be moved.</td>
<td></td>
</tr>
<tr>
<td>Application Extensible Flexfield</td>
<td>Extensible flexfield registration data and setup data, including categories</td>
<td>No parameters: all extensible flexfields are moved moduleType/moduleKey: only extensible flexfields belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved. extensibleFlexfieldCode/applicationKeyId: only the specified extensible flexfield is moved.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Only flexfields with a deployment status of Deployed or Deployed to Sandbox are eligible to be moved.</td>
<td></td>
</tr>
</tbody>
</table>
| Application Key Flexfield | Key flexfield registration data and setup data | No parameters: all key flexfields are moved.  

moduleType/moduleKey: only key flexfields belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.  

keyFlexfieldCode/applicationId: only the specified key flexfield is moved.  

**Note**  
Importing a flexfield’s metadata can change its deployment status and therefore, the affected flexfields must be redeployed. The import process automatically submits affected flexfields for redeployment.  

**Note**  
Only flexfields with a deployment status of Deployed or Deployed to Sandbox are eligible to be moved. |
| Application Flexfield Value Set | Value set setup data | No parameters: all value sets are moved.  

moduleType/moduleKey: only value sets belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.  

valueSetCode: only the specified value set is moved.  

**Note**  
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>| 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. |</p>
<table>
<thead>
<tr>
<th>Application Reference Language</th>
<th>Language data</th>
<th>No parameters: all languages are moved.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Reference Natural Language</td>
<td>Natural language data</td>
<td>No parameters: all natural languages are moved.</td>
</tr>
<tr>
<td>Application Reference Territory</td>
<td>Territory data</td>
<td>No parameters: all territories are moved.</td>
</tr>
<tr>
<td>Application Reference Time zone</td>
<td>Time zone data</td>
<td>No parameters: all time zones are moved.</td>
</tr>
<tr>
<td>Application Standard Lookup</td>
<td>Standard lookup types and their lookup codes</td>
<td>No parameters: all standard lookups are moved. moduleType/moduleKey: only standard lookups belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved. lookupType: only the specified common lookup is moved.</td>
</tr>
<tr>
<td>Application Common Lookup</td>
<td>Common lookup types and their lookup codes</td>
<td>No parameters: all common lookups are moved. moduleType/moduleKey - only common lookups belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved. lookupType: only the specified common lookup is moved.</td>
</tr>
<tr>
<td>Application Set-Enabled Lookup</td>
<td>Set-enabled lookup types and their lookup codes</td>
<td>No parameters: all set-enabled lookups are moved. moduleType/moduleKey: only set-enabled lookups belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved. lookupType: only the specified set-enabled lookup is moved.</td>
</tr>
<tr>
<td>Application Profile Category</td>
<td>Profile categories</td>
<td>No parameters: all profile categories are moved. moduleType/moduleKey: only categories belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved. name/applicationId: only the specified category is moved.</td>
</tr>
</tbody>
</table>
| Application Profile Option | Profile options and their values | No parameters: all profile options and their values are moved.  
|                           |                                | moduleType/moduleKey: only profile options and their values belonging to the specified module are moved.  
|                           |                                | profileOptionName: only the specified profile option and its values are moved.  
| Application Profile Value | Profile options and their values | No parameters: all profiles and their values are moved.  
|                           |                                | moduleType/moduleKey: only profiles and their values belonging to the specified module are moved.  
|                           |                                | categoryName/categoryApplicationId: only profiles and their values belonging to the specified category are moved.  
|                           |                                | profileOptionName: only the specified profile and its values are moved.  
| Application Reference Data Set | Reference data sets | No parameters: all sets are moved.  
| Application Reference Data Set Assignment | Reference data set assignments | determinantType: only assignments for the specified determinant type are moved.  
|                                           |                                | determinantType/referenceGroupName: only assignments for the specified determinant type and reference group are moved.  
| Application Tree Structure | Tree structures and any labels assigned to the tree structure | No parameters: all tree structures (and their labels) are moved.  
|                                           |                                | moduleType/moduleKey: only tree structures (and their labels) belonging to the specified module are moved.  
|                                           |                                | treeStructureCode: only the specified tree structure (with its labels) is moved.  

| Application Tree | Tree codes and versions | No parameters: all trees are moved.  
moduleType/moduleKey: only trees belonging to the specified module are moved.  
treeStructureCode: only trees belonging to the specified tree structure are moved.  
TreeStructureCode/TreeCode: only trees belonging to the specified tree structure and tree code are moved. |
| Application Tree Label | Tree structures and any labels assigned to the tree structure | No parameters: all tree structures (and their labels) are moved.  
moduleType/moduleKey: only tree structures (and their labels) belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.  
treeStructureCode: only the specified tree structure (with its labels) is moved. |
| Application Data Security Policy | Database resources, actions, conditions, and data security policies | No parameters: all database resources/actions/conditions/policies are moved.  
moduleType/moduleKey: only database resources/actions/conditions/policies belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.  
objName: only the specified database resource along with its actions/conditions/policies is moved. |
| **Note** | |  
- If the policies being moved contain reference to newly created roles, move the roles before moving the policies.  
- If the source and target systems use different LDAPs, manually perform the GUID reconciliation after moving the data security policies. |
| 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.
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.

accounting method
A set of journal entry rules which determine how a subledger journal entry is to be created for each event class or event type.

action
The kind of access named in a security policy, such as view or edit.

ADF
Acronym for Application Developer Framework. A set of programming principles and rules for developing software applications.

application feature
A standardized functionality that is available to implemented.

application identity
Predefined application level user with elevated privileges. An application identity authorizes jobs and transactions for which other users are not authorized, such as a payroll run authorized to access a taxpayer ID while the user who initiated the job is not authorized to access such personally identifiable information.

application role
A role specific to applications and stored in the policy store.

Applications Core
Abbreviation for Oracle Middleware Extensions for Applications. The technical product code is FND.

assignment
A set of information, including job, position, pay, compensation, managers, working hours, and work location, that defines a worker’s or nonworker’s role in a legal employer.
balancing segment
A chart of accounts segment used to automatically balance all journal entries for each value of this segment.

BPEL
Business Process Execution Language; a standard language for defining how to send XML messages to remote services, manipulate XML data structures, receive XML messages asynchronously from remote services, manage events and exceptions, define parallel sequences of execution, and undo parts of processes when exceptions occur.

business function
A business process, or an activity that can be performed by people working within a business unit and describes how a business unit is used.

business object
A resource in an enterprise database, such as an invoice or purchase order.

business unit
A unit of an enterprise that performs one or many business functions that can be rolled up in a management hierarchy.

calendar event
A period that signifies an event, such as a public holiday or a training course, that impacts worker availability.

catalog
A collection of categories used to classify items which can be organized into a hierarchy that represents a taxonomy.

chart of accounts
The account structure your organization uses to record transactions and maintain account balances.

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.

corporate rate type
Rate you define to standardize rates used in conversion of one currency to another over a period of time. This rate is generally a standard market rate determined by senior financial management for use throughout the organization.

cost center
A unit of activity or group of employees used to assign costs for accounting purposes.

cost organization
A grouping of inventory organizations that indicates legal and financial ownership of inventory, and which establishes common costing and accounting policies.

country holding company
A legal entity that acts on behalf of several divisions within an enterprise, and is the legal employer in a country.

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.

**descriptive flexfield**
An extendable field that captures additional information.

**determinant**
A value that determines which reference data set will be used in a specific business context.

**determinant type**
Designates the field within transactional columns that controls how data is shared across organizations such as business unit, asset book, cost organization or project unit. The type determines the reference data sets that would be used in a transaction.

**determinant type**
An additional and optional field within transactional columns (besides category and application) that is used to assign document sequences. The available determinant types are Business Unit, Ledger, Legal Entity, and Tax Registration.

**determinant value**
A value specific to the determinant type dimension of a document sequence. The determinant value is relevant in a document sequence assignment only if the document sequence has a determinant type. If Ledger is the determinant type for a document sequence, the determinant value is the specific ledger number whose documents are numbered by the document sequence.

**division**
A business-oriented subdivision within an enterprise. Each division is organized to deliver products and services or address different markets.
**document sequence**
A unique number that is automatically or manually assigned to a created and saved document.

**duty role**
A group of function and data privileges representing one duty of a job. Duty roles are specific to applications, stored in the policy store, and shared within an Oracle Fusion Applications instance.

**enterprise**
An organization with one or more legal entities under common control.

**enterprise role**
Abstract, job, and data roles are shared across the enterprise. An enterprise role is an LDAP group. An enterprise role is propagated and synchronized across Oracle Fusion Middleware, where it is considered to be an external role or role not specifically defined within applications.

**entitlement**
Grants of access to functions and data. Oracle Fusion Middleware term for privilege.

**ESS**
Acronym for Enterprise Storage Server. An application that optimizes data storage.

**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 let you group contexts into categories.

**feature choice**
A selection you make when configuring offerings that modifies a setup task list, or a setup page, or both.

**fixed rate type**
Rate you set between two currencies that remains constant. For example, a rate set between the euro currency and each Economic and Monetary Union (EMU) currency during the conversion to the euro currency.

**flexfield**
Grouping of extensible data fields called segments, where each segment is an attribute added to an entity for capturing additional information.
**flexfield segment**
An extensible data field that represents an attribute on an entity and captures a single atomic value corresponding to a predefined, single extension column in the Oracle Fusion Applications database. A segment appears globally or based on a context of other captured information.

**function security**
The control of access to a page or a specific widget or functionality within a page. Function security controls what a user can do.

**global area**
The region across the top of the user interface. It provides access to features and tools that are relevant to any page you are on.

**grade**
A component of the employment model that defines the level of compensation for a worker.

**GTIN**
Abbreviation for Global Trade Identification Number

**HCM**
Abbreviation for Human Capital Management.

**HCM data role**
A job role, such as benefits administrator, associated with specified instances of Oracle Fusion Human Capital Management (HCM) data, such as one or more positions or all persons in a department.

**identity**
A person representing a worker, supplier, or customer.

**import**
In the context of data integration, the transfer of data from interface tables to application tables, where the data is available to application users.

**interface table**
A database table used for transferring data between applications or from an external application or data file.

**item master**
A collection of data that describes items and their attributes recorded in a database file.
item organization
Item definition where inventory balances are not stored and movement of inventory is not tracked in the applications. Item attributes that carry financial and accounting information are hidden.

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.

**load**

In the context of data integration, the transfer of external data from data files to the receiving *interface tables* in preparation for an import into application tables.

**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.

**natural account**

Categorizes account segment values by account type, asset, liability, expense, revenue, or equity, and sets posting, budgeting, and other options.

**natural account segment**

A chart of accounts segment used to categorize your accounting transactions by account type: asset, liability, owner's equity, revenue, or expense.

**offering**

A comprehensive grouping of business functions, such as Sales or Product Management, that is delivered as a unit to support one or more business processes.

**OWLCS**

Abbreviation for Oracle WebLogic Communication Services. Offers the TPCC service to Oracle Sales Cloud and sets up the calls via SIP integration with the telephony network.
party
A physical entity, such as a person, organization or group, that the deploying company has an interest in tracking.

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.

person number
A person ID that is unique in the enterprise, allocated automatically or manually, and valid throughout the enterprise for all of a person's work and person-to-person relationships.

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.

profile option
User preferences and system configuration options consisting of a name and a value, that can be set at hierarchical levels of an enterprise. Also called a profile or user option.

profile option level
A level at which profile option values are defined. Site, product, and user are predefined levels.
**profile option level hierarchy**
The ordering of profile option levels. The order of the levels in the hierarchy determines which levels take precedence.

**profile option value**
The value portion of a profile option's name and value. A profile option may have multiple values set at different levels, such as site or user.

**PSTN**
Abbreviation for public switched telephone network which is the network of the world’s public circuit-switched telephone networks.

**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 run time 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, and payroll flows.

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.

semantic model
A fully configured data lens constructed in Oracle Product Data Quality, which provides the basis for checking data quality. The semantic model contains your definitions for classification, standardization, and matching.

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.

spot rate type
Rate you enter to perform conversion based on this rate as of a specific date. This rate applies to the immediate delivery of a currency.

SQL predicate
A type of condition using SQL to constrain the data secured by a data security policy.
**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.

**user rate type**

Rate you enter at journal entry time to convert foreign currency transactions to your functional currency.

**value set**

A set of valid values against which values entered by an end user are validated. The set may be tree structured (hierarchical).

**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.