

**Oracle® Fusion Applications Order Orchestration
Implementation Guide**

11g Release 7 (11.1.7)

Part Number E20386-07

March 2013

Oracle® Fusion Applications Order Orchestration Implementation Guide

Part Number E20386-07

Copyright © 2011-2013, Oracle and/or its affiliates. All rights reserved.

Authors: Cynthia Stevens, Deborah West, Tina Brand, Essan Ni Jirman, Suzanne Kinkead, P. S. G. V. Sekhar, Barbara Snyder, Srinivas Vellikad, Megan Wallace, Kathryn Wohnoutka, Jacqueline Wood, Brent Bosin, Douglas Myers

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

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

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

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

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

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

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

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

Contents

1 Overview

Order Orchestration 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 Define Implementation Users

Initial Security Administration: Critical Choices	3-1
---	-----

4 Define Enterprise Structures for Order Orchestration

Enterprise Structures: Overview	4-1
Enterprise Structures Business Process Model: Explained	4-3
Global Enterprise Configuration: Points to Consider	4-5
Modeling Your Enterprise Management Structure in Oracle Fusion: Example	4-6
Define Initial Configuration with the Enterprise Structures Configurator	4-9
Define Reference Data Sharing	4-33
Define Geographies	4-41
Define Enterprise for Order Orchestration: Manage Enterprise HCM Information	4-62
Define Enterprise for Order Orchestration: Manage Locations	4-63
Define Business Units: Manage Business Units	4-65
Define Facilities for Order Orchestration	4-66

5 Define Security for Supply Chain Management

Security Tasks: Highlights	5-1
Defining Security After Enterprise Setup: Points to Consider	5-4
Security Tasks and Oracle Fusion Applications: How They Fit Together	5-8
Security Tasks: Overview	5-11
Define Data Security for Supply Chain Management	5-15
Define Users for Supply Chain Management	5-34

6 Define Automated Governance, Risk, and Performance Controls

Segregation of Duties: Explained	6-1
--	-----

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

7 Define Help Configuration

Define Help Configuration: Overview	7-1
Set Help Options	7-1
FAQs for Assign Help Text Administration Duty	7-4
Manage Help Security Groups	7-4
Help File Customization	7-5
Embedded Help Customization	7-16

8 Maintain Common Reference Objects

Maintain Common Reference Objects: Overview	8-1
Define Application Taxonomy	8-1
Define Reference Data Sharing	8-4
Define Lookups	8-7
Define ISO Reference Data	8-14
Manage Data Security Policies	8-18
Define Document Sequences	8-28
Define Trees	8-31
Define Profile Options	8-46
Define Flexfields	8-55
Define Attachments	8-98
Set Activity Stream Options	8-102
Manage Menu Customizations	8-102
Manage Audit Policies	8-103
Manage Oracle Social Network Objects	8-105
Manage Applications Core Common Reference Objects: Manage Applications Core Messages	8-109
Manage Applications Core Common Reference Objects: Manage Applications Core Administrator Profile Values	8-109

9 Define WebLogic Communication Services Configuration

Click-to-Dial: Top Tasks	9-1
Enhanced Click-to-Dial Setup: Explained	9-2
Enhanced Click-to-Dial Components: How They Work Together	9-6

10 Define Common Order Orchestration Configuration

Item Imports: Explained	10-1
Define Catalogs: Create Catalogs	10-2
Define Catalogs: Manage Catalogs	10-6
Manage Units of Measure	10-21

Manage Customers: Define Source Systems	10-23
Manage Customers: Import Person and Organization	10-25

11 Define Application Toolkit Configuration

Define Application Toolkit Configuration: Overview	11-1
Map Reports to Work Areas	11-1
Set Watchlist Options	11-3

12 Define Order Promising and Perform Data Collections

Manage Planning Source Systems	12-1
Data Collections, Order Orchestration, and Order Promising: How They Fit Together	12-5
Collecting Data for the Order Orchestration and Planning Data Repository: Explained	12-7
Data Collection Entities: Explained	12-10
Collect Order Promising Reference and Transaction Data	12-13
Manage Sourcing Rules and Bills of Distribution	12-35
Manage Assignment Sets	12-40
Manage Global Order Promising Profile Options	12-47

13 Manage Order Promising Rules

ATP Rules, Allocation Rules, and Sourcing Rules: How They Work Together	13-1
Manage ATP Rules	13-2
Manage Planning Allocation Rules	13-11

14 Define Sales Order Fulfillment

Oracle Fusion Distributed Order Orchestration Components: How They Work Together	14-1
Orchestration Lookups: Explained	14-2
Orchestration Profile Management: Points to Consider	14-3
Oracle Fusion Distributed Order Orchestration Extensible Flexfields: Explained	14-4
Oracle Fusion Distributed Order Orchestration Extensible Flexfield Uses: Explained	14-4
Oracle Fusion Distributed Order Orchestration Extensible Flexfield Setup: Explained	14-6
Manage Hold Codes	14-7
Manage Orchestration Source Systems	14-8
Collect Orchestration Reference and Transaction Data	14-12
Define Orchestration	14-40
Define Processing Constraints	14-87
Define Transformation Details	14-90

15 Define Transactional Business Intelligence Configuration

Define Transactional Business Intelligence Configuration: Highlights	15-1
Access to Person Data	15-1

16 Define Extensions for Order Orchestration

Define Custom Enterprise Scheduler Jobs for Order Orchestration	16-1
---	------

17 Importing and Exporting Setup Data

Configuration Packages: Explained	17-1
Exporting and Importing Setup Data: Explained	17-1
Moving Common Reference Objects	17-2

18 External Integration

Web Services: Overview	18-1
Files for Import and Export	18-2
External Data Integration Services for Oracle Cloud	18-6

Preface

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

Oracle Fusion Applications Help

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



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

Oracle Fusion Applications Guides

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

Guides are designed for specific audiences:

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

intended for implementors, security administrators, and auditors. They are organized by role.

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

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

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

Other Information Sources

My Oracle Support

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

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

Oracle Enterprise Repository for Oracle Fusion Applications

Oracle Enterprise Repository for Oracle Fusion Applications provides 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.

Note

The content of Oracle Enterprise Repository reflects the latest release of Oracle Fusion Applications.

Documentation Accessibility

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

Comments and Suggestions

Your comments are important to us. We encourage you to send us feedback about Oracle Fusion Applications Help and guides. Please send your suggestions to oracle_fusion_applications_help_ww_grp@oracle.com. You can use the **Send Feedback to Oracle** link in the footer of Oracle Fusion Applications Help.

Overview

Order Orchestration Offering: Overview

By implementing the Order Orchestration offering, your enterprise can set up order fulfillment to reflect operating practices, improve effectiveness and consistency of business processes, and enforce corporate compliance.

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 Order Orchestration offering, you can select the following options:

- Order Orchestration (this option also has associated features that you can select to configure)
- Supply Chain and Order Management Business Intelligence Analytics
- Order Management Business Intelligence Analytics
- Logistics Business Intelligence Analytics

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

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

- Define Common Applications Configuration for Order Orchestration
- Define Common Order Orchestration Configuration
- Define Order Promising
- Collect Order Promising Reference and Transaction Data

- Manage Order Promising Rules
- Define Sales Order Fulfillment
- Define Transactional Business Intelligence Configuration
- Define Extensions for Order Orchestration

Define Common Applications Configuration for Order Orchestration

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 Common Order Orchestration Configuration

Use this task list to define the configuration for common setup, such as units of measure and catalogs for Oracle Fusion Distributed Order Orchestration.

Define Order Promising

Use this task list to enable order promising by defining specific supply sources and the data that needs to be collected from each source, and assigning sourcing rules or rule sets to items in order to select the correct source organization from one of the fulfillment source systems in your supply chain.

Collect Order Promising Reference and Transaction Data

Use this task list to define collection of supply data from multiple sources, including inventory and planning systems, and to define supply update frequency across supply sources.

Manage Order Promising Rules

Use this task list to define order promising rules for items by customer and location, the method used to allocate supply across classes of demand, and the priority and sequence of supply sources to check for supply when there are multiple sources for an item.

Define Sales Order Fulfillment

Use this task list to configure decomposition rules, orchestration processes, planning and jeopardy conditions, and change order logic. You can also define holds, constraints, and status, and register new services and integration targets.

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 Order Orchestration

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

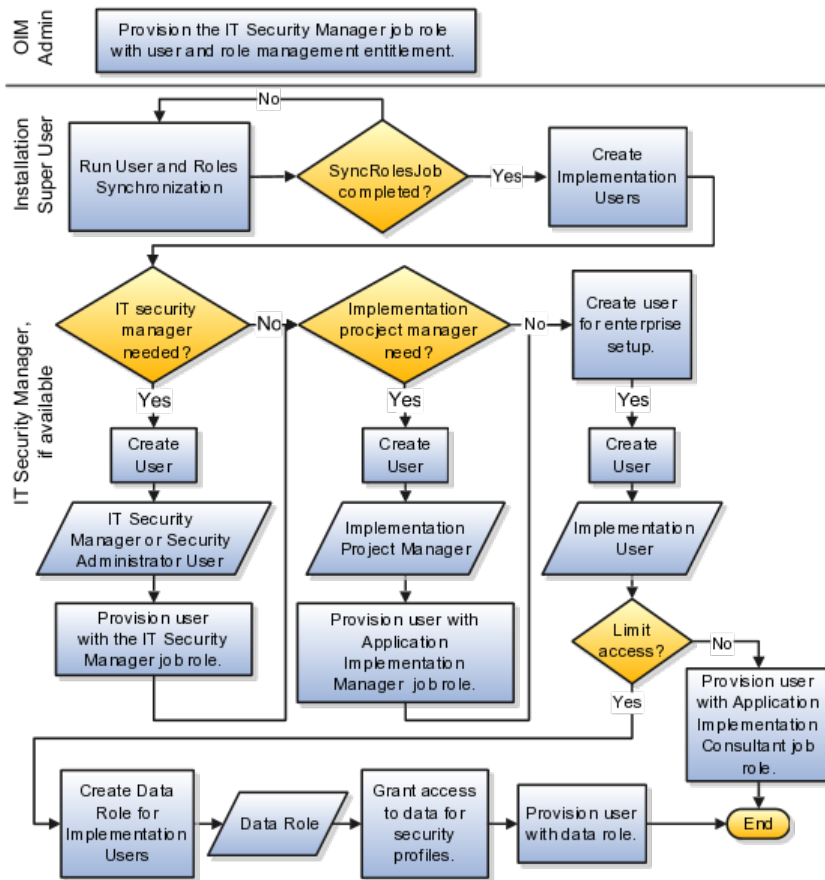
To start an Oracle Fusion Applications implementation, you must set up one or more initial users using the super user that was created during installation and provisioning of the Oracle Fusion Applications environment, or using the initial administrator user provided by Oracle for Oracle Cloud Application 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:
 - a. Synchronize users and roles in the Lightweight Directory Access Protocol (LDAP) store with HCM user management by using the Run User and Roles Synchronization Process task.
 - b. Create an IT security manager user by using the Create Implementation Users task.
 - c. Provision the IT security manager with the IT Security Manager role by using the Provision Roles to Implementation Users task.
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. The Application Implementation Consultant job role inherits from all product-specific application administrators and entitles the necessary View All access to all secured object.
 - c. Optionally, create a data role for an implementation user who needs only the limited access of a product-specific Application Administrator

by using the Create Data Role for Implementation Users. Then assign the resulting data role to the implementation user by using the Provision Roles to Implementation Users task.

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



Manage Application Implementation

Manage Application Implementation: Overview

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

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

- **Prepackaged Lists of Implementation Tasks**

Task lists can be easily configured and extended to better fit with business requirements. Auto-generated, sequential task lists include prerequisites

and address dependencies to give full visibility to end-to-end setup requirements of Oracle Fusion applications.

- **Rapid Start**

Specific implementations can become templates to facilitate reuse and rapid-start 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

Implementation Projects: Explained

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

Maintaining Setup Data

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

Exporting and Importing

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

Selecting Offerings

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

Offerings: Explained

Offerings are application solution sets representing one or more business processes and activities that you typically provision and implement as a unit. They are, therefore, the primary drivers of functional setup of Oracle Fusion applications. Some of the examples of offerings are Financials, Procurement,

Sales, Marketing, Order Orchestration, and Workforce Deployment. An offering may have one or more options or feature choices.

Implementation Task Lists

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

Enabling Offerings

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

Provisioning Offerings

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

Options: Explained

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

Feature Choices: Explained

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

You can configure feature choices in three different ways:

Yes or No

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

Single Select

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

Multi-Select

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

Define Synchronization of Users and Roles from LDAP

User and Role Synchronization: Explained

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

Note

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

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

Tip

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

Define Implementation Users

Initial Security Administration: Critical Choices

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

Initial security administration consists of the following.

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

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

Preparing the IT Security Manager Job Role

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

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

- Identity User Administrators
- Role Administrators

Additionally, you must assign the Xellerate Users organization to the IT Security Manager role.

Synchronizing Users and Roles from LDAP

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

Defining Initial Implementation Users

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

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

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

Creating Implementation Users

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

Note

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

Creating Data Roles for Implementation Users

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

Provisioning Roles to Implementation Users

After creating an implementation user, you must provision the user with one or more roles by performing the Provision Roles to Implementation Users task.

For example, assign a role to the implementation user that provides the access necessary for setting up the enterprise. Depending on need, provision to the implementation user the predefined Applications Implementation Consultant role or a product family-specific administrator data role, such as a data role based on the predefined Financials Applications Administrator.

Caution

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

Initial Security Administration: Worked Example

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

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

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

The following table summarizes key decisions for this scenario.

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

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

1. The Oracle Identity Management System Administrator user provisions the IT Security Manager job role with roles for user and role management.
2. The Oracle Fusion Applications super user synchronizes LDAP users with HCM user management so that users can be provisioned with roles through HCM.
3. The Oracle Fusion Applications super user performs the Create Implementation Users task to create one or more IT security manager and administrator users provisioned with security administrative entitlement.
4. The IT Security Manager user signs in to Oracle Fusion Applications and performs the Create Implementation Users task to create implementation managers and users.
5. The IT Security Manager user provisions implementation users for enterprise structure setup.

Note

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

Preparing the IT Security Manager Role

The super user that was created when installing and provisioning Oracle Fusion Applications (for example, FAADMIN), 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:

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

Note

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

4. Click **Save**.
5. On the Roles tab in the IT_SECURITY_MANAGER user creation task flow, click **Assign**.
6. In the Add Role window, search for the IT Security Manager role and click **Add**.

Defining an Implementation User for Enterprise Structures Setup

1. Sign in to Oracle Fusion Applications using the IT security manager user's name and password.
2. Create and provision an implementation user using the same task flow as for creating the IT security manager user in the previous section, except provision the following roles.
 - Application Implementation Manager

- Application Implementation Consultant

Note

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

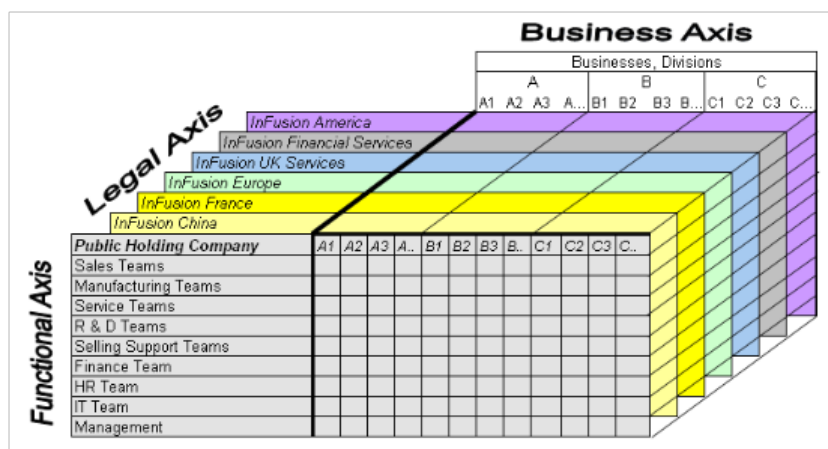
Define Enterprise Structures for Order Orchestration

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

employment comes from your charter in the legal system. A corporation is a distinct legal entity from its owners and managers. The corporation is owned by its shareholders, who may be individuals or other corporations. There are many other kinds of legal entities, such as sole proprietorships, partnerships, and government agencies.

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

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

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

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

Management Structure

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

Functional Structure

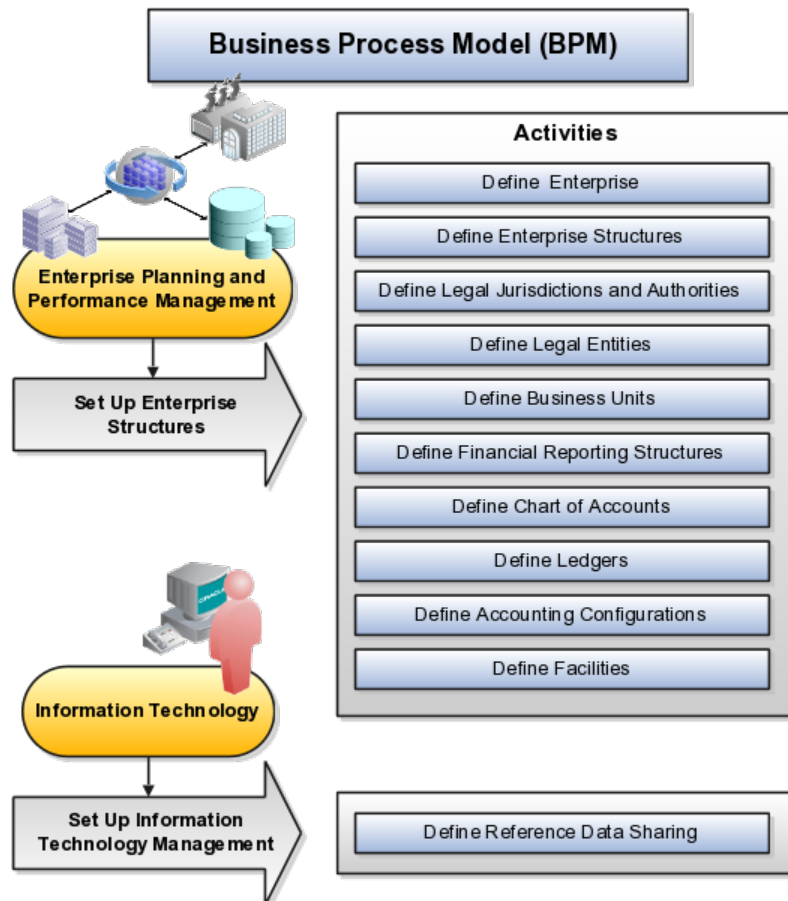
Straddling the legal and business organizations is a functional organization structured around people and their competencies. For example, sales, manufacturing, and service teams are functional organizations. This functional structure is represented by the Functional Axis in the figure above. You reflect

the efforts and expenses of your functional organizations directly on the income statement. Organizations must manage and report revenues, cost of sales, and functional expenses such as research and development (R&D) and selling, general, and administrative (SG&A) expenses. In Oracle Fusion Applications, the functional structure is implemented using departments and organizations, including sales, marketing, project, cost, and inventory organizations.

Enterprise Structures Business Process Model: Explained

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

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



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

Note

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

Global Enterprise Configuration: Points to Consider

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

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

Enterprise Configuration

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

Business Unit Management

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

Security Structure

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

Compliance Requirements

How do you comply with your corporate external reporting requirements and local statutory reporting requirements? Do you tend to prefer a corporate first or

an autonomous local approach? Where are you on a spectrum of centralization, very centralized or decentralized?

Modeling Your Enterprise Management Structure in Oracle Fusion: Example

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

Scenario

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

InFusion Corporation

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

Analysis

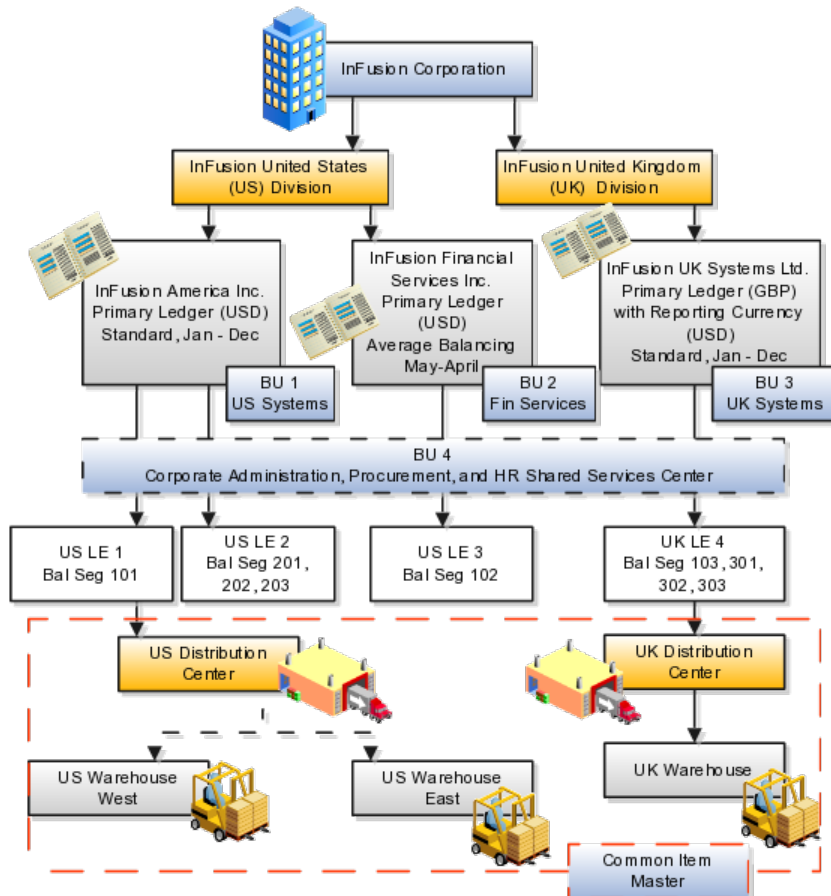
The following are elements you need to consider in creating your model for your global enterprise structure.

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

Global Enterprise Structure Model

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

- Creation of three separate ledgers representing your separate legal entities:
 - InFusion America Inc.
 - InFusion Financial Services Inc.
 - InFusion UK Services Ltd.
- Consolidation of results for system components, installations, and maintenance product lines across the enterprise
- All UK general and administrative costs processed at the UK headquarters
- US Systems' general and administrative costs processed at US Corporate headquarters
- US Financial Services maintains its own payables and receivables departments



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

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

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

Note

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

Note

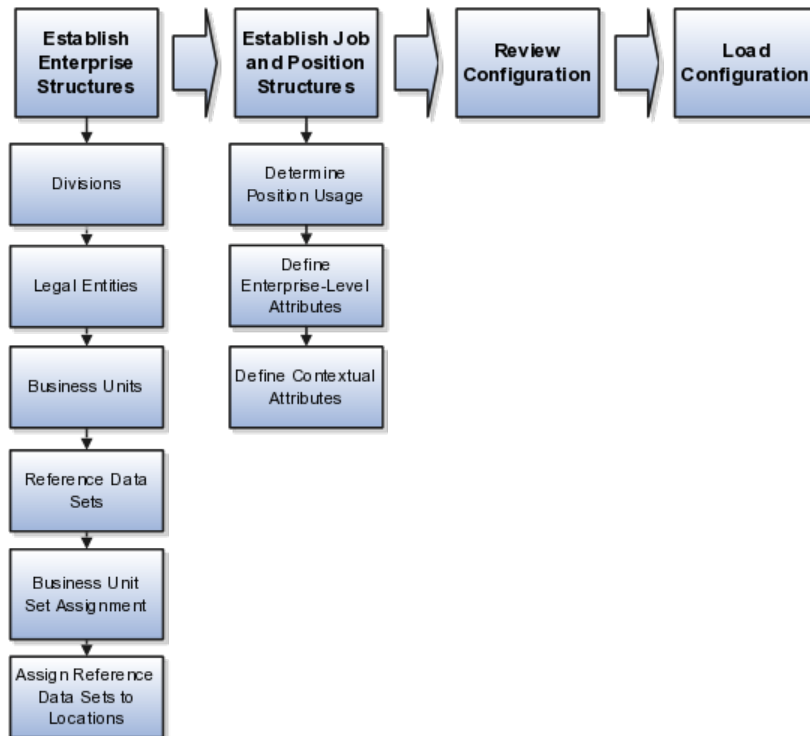
The InFusion Corporation is a legal entity but is not discussed in this example.

Define Initial Configuration with the Enterprise Structures Configurator

Establishing Enterprise Structures Using the Enterprise Structures Configurator: Explained

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

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



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

Establish Enterprise Structures

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

Establish Job and Position Structures

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

Review Configuration

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

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

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

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

Note

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

Load Configuration

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

Using Rollback for an Enterprise Structures Configuration: Explained

The Enterprise Structures Configurator provides the ability to roll back, or undo, an enterprise configuration. Two methods for rolling back a configuration are available: manual rollback, and automatic rollback.

Manual Rollback

Use the manual method for rolling back an enterprise configuration when you have loaded a configuration, but then decide you do not want to use it.

Automatic Rollback

The automatic rollback is used when you run the Load Configuration process, but the process encounters an error. In this case, the application 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.

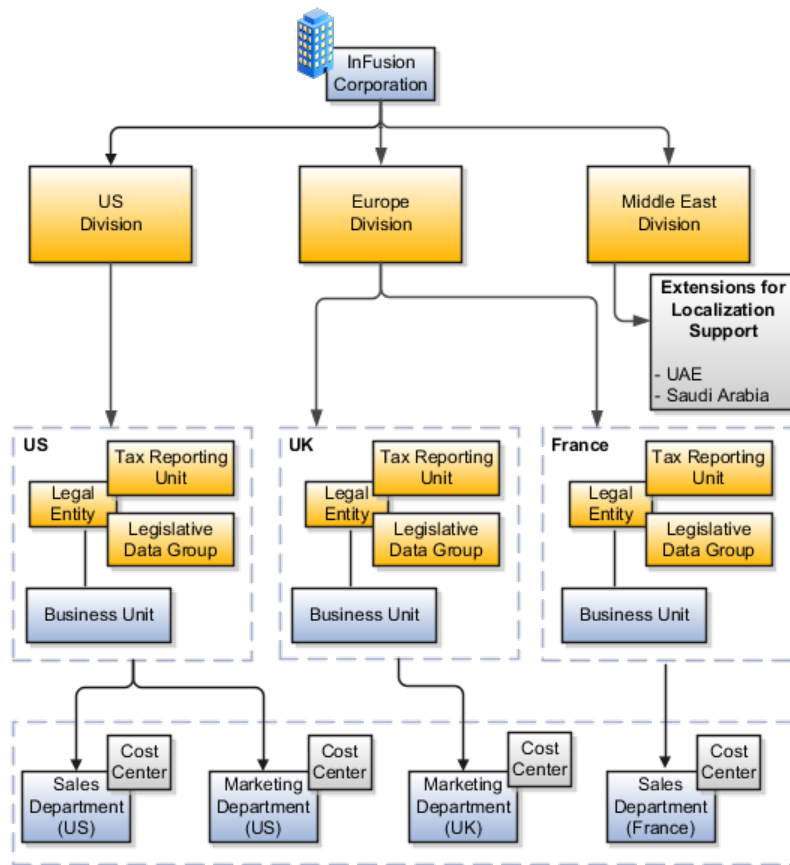
InFusion requires legal entities with legal employers, payroll statutory units, tax reporting units, and legislative data groups for the US, UK, France, Saudi Arabia, and UAE, in order to employ and pay its workers in those countries.

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.

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

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



Division: Explained

Managing multiple businesses requires that you segregate them by their strategic objectives and measure their results. Responsibility to reach objectives can be delegated along the management structure. Although related to your legal structure, the business organizational hierarchies do not need to reflect directly

the legal structure of the enterprise. The management entities and structure can include divisions and subdivisions, lines of business, and other strategic business units, and include their own revenue and cost centers. These organizations can be included in many alternative hierarchies and used for reporting, as long as they have representation in the chart of accounts.

Divisions

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

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

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

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

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

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

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

Legal Entities: Explained

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

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

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

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

The Role of Your Legal Entities

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

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

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

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

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

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

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

Automatically Creating Legal Entities

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

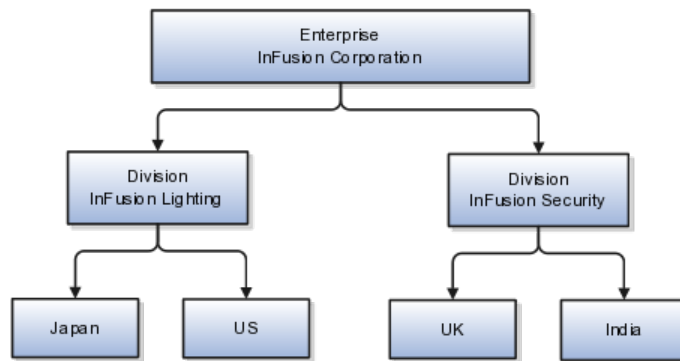
The application automatically creates the legal entities that you select, and identifies them as payroll statutory units and legal employers. For each country that you indicated that your enterprise operates in, and for each country that you created a location for, the application also automatically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

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

Example: Creating Legal Entities Automatically

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

This figure illustrates InFusion Corporation's enterprise structure.



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

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

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

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

Creating Legal Entities Using a Spreadsheet

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

Legal Entity in Oracle Fusion: Points to Consider

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

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

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

The Importance of Legal Entity in Transactions

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

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

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

Legal Entity and Its Relationship to Business Units

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

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

Legal Entity and Its Relationship to Divisions

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

Legal Entity and Its Relationship to Ledgers

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

Legal Entity and Its Relationship to Balancing Segments

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

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

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

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

Note

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

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

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

Note

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

Legal Entity and Its Relationship to Consolidation Rules

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

Legal Entity and its Relationship to Intercompany Transactions

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

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

Note

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

Legal Entity and Its Relationship to Worker Assignments and Legal Employer

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

Legal Entity and Payroll Reporting

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

Business Units: Explained

A business unit is a unit of an enterprise that performs one or many business functions that can be rolled up in a management hierarchy. A business unit can

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

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

The Oracle Fusion Applications business unit model:

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

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

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

In summary, use business units in the following ways:

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

Brief Overview of Business Unit Security

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

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

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

Creating Business Units in the Enterprise Structures Configurator: Points to Consider

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

Automatically Creating Business Units

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

You can automatically create business units at the following levels:

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

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

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

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

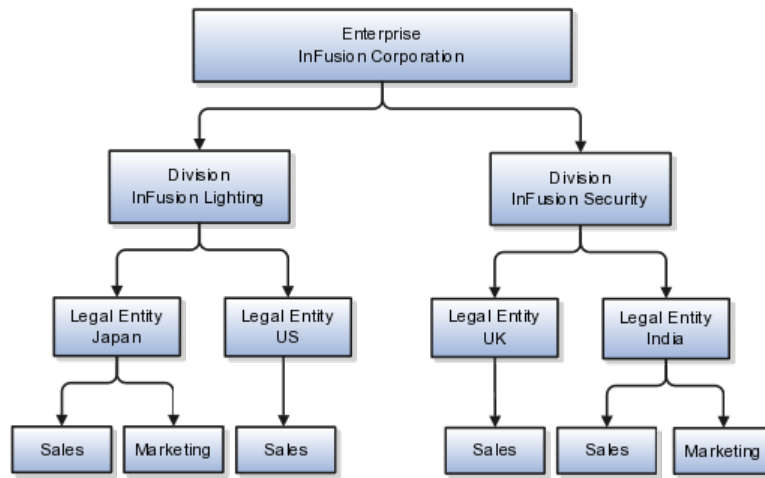
The business units listed by the application are suggestions only, and are meant to simplify the process to create business units. You are not required to select all of the business units suggested. When you navigate to the next page in the ESC guided flow, which is the Manage Business Units page, you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

the business units that were created automatically. You must return to the Create Business Units page and deselect any business units that you no longer want.

Example: Selecting Business Unit Levels

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

This figure illustrates InFusion Corporation's enterprise structure.



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

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

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

Manually Creating Business Units

If none of the levels for creating business units meets your business needs, you can create business units manually, and you create them on the Manage Business Units page. If you create business units manually, then no reference data sets are created automatically. You must create them manually as well.

Reference Data Sets and Sharing Methods: Explained

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

The reference data sharing features use reference data sets to which reference data is assigned. The reference data sets group assigned reference data. The sets

can be understood as buckets of reference data assigned to multiple business units or other application components.

Reference Data Sets

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

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

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

Reference Data Sharing Methods

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

- Assignment to one set only, no common values allowed. The simplest form of sharing reference data that allows assigning a reference data object instance to one and only one set. For example, Asset Prorate Conventions are defined and assigned to only one reference data set. This set can be shared across multiple asset books, but all the values are contained only in this one set.
- Assignment to one set only, with common values. The most commonly used method of sharing reference data that allows defining reference data object instance across all sets. For example, Receivables Transaction Types are assigned to a common set that is available to all the business units without the need to be explicitly assigned the transaction types to each business unit. In addition, you can assign a business unit specific set of transaction types. At transaction entry, the list of values for transaction types includes transaction types from the set assigned to the business unit, as well as transaction types assigned to the common set that is shared across all business units.
- Assignment to multiple sets, no common values allowed. The method of sharing reference data that allows a reference data object instance to be assigned to multiple sets. For instance, Payables Payment Terms use this method. It means that each payment term can be assigned to one or more than one set. For example, you assign the payment term Net 30 to several sets, but the payment term Net 15 is assigned to only your corporate business unit specific set. At transaction entry, the list of values

for payment terms consists of only one set of data; the set that is assigned to the transaction's business unit.

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

Business Units and Reference Data Sets: How They Work Together

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

Common Set Versus Specific Sets

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

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

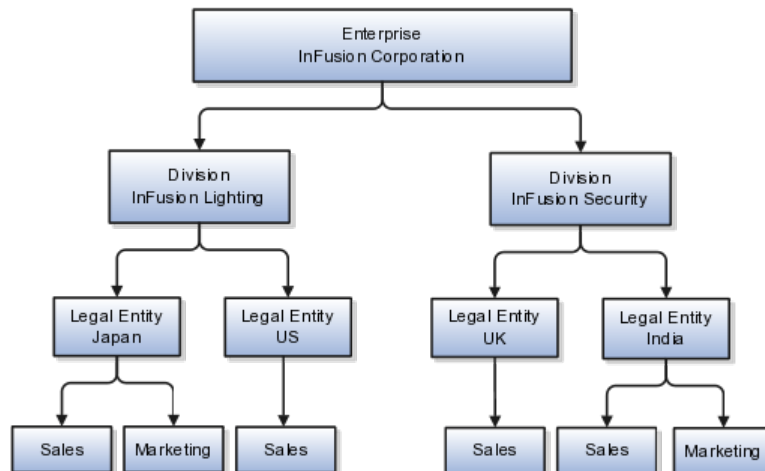
Business Unit Set Assignment

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

Example: Assigning Sets to Business Units

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

The following figure illustrates the structure of InFusion Corporation.



When deciding how to create business units, InFusion decides to create them using the country and business function level. Therefore, they created the following business units:

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

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

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

InFusion assigns business units to sets as follows:

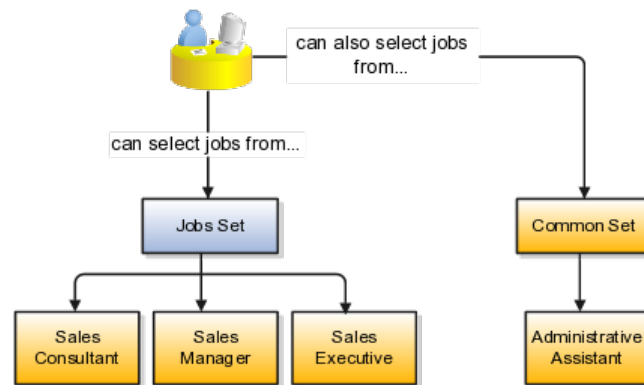
Business Unit	Default Set Assignment	Set Assignment Overrides
Sales_Japan	Sales_Japan_Set for grades, departments, and locations	Jobs set for jobs
Marketing_Japan	Mktg_Japan_Set for grades, departments, and locations	None
Sales_US	Sales_US_Set for grades, departments, and locations	Jobs set for jobs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

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

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

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

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



Creating Reference Data Sets in the Enterprise Structures Configurator: Explained

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

A standard set called the Enterprise set is predefined.

Common Set

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

Jobs and Positions: Critical Choices

Jobs and positions represent roles that enable you to distinguish between tasks and the individuals who perform those tasks. The key to whether to use jobs or

positions is how each is used. Positions offer a well-defined space independent of the person performing the job. Jobs are a space defined by the person. A job can be defined globally in the Common Set, whereas a position is defined within one business unit.

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

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

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

Primary Industry of Your Enterprise

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

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

Management of People

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

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

Positions: Examples

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

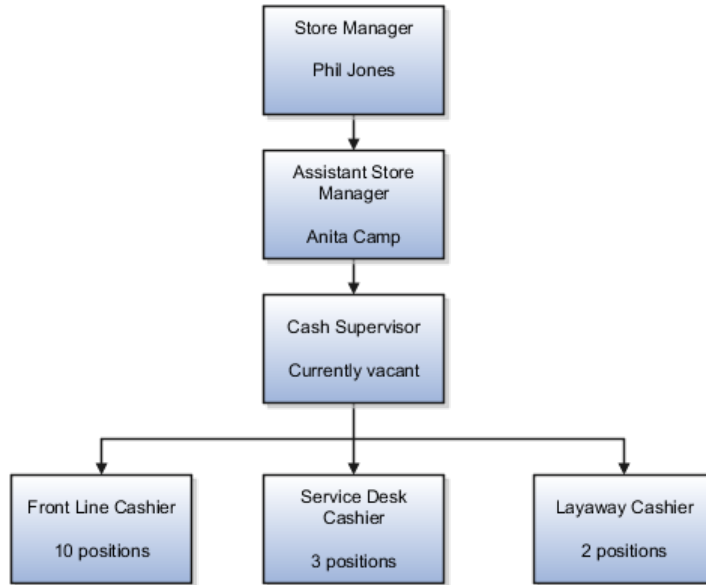
Retail Industry

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

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

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

This figure illustrates the retail position setup.

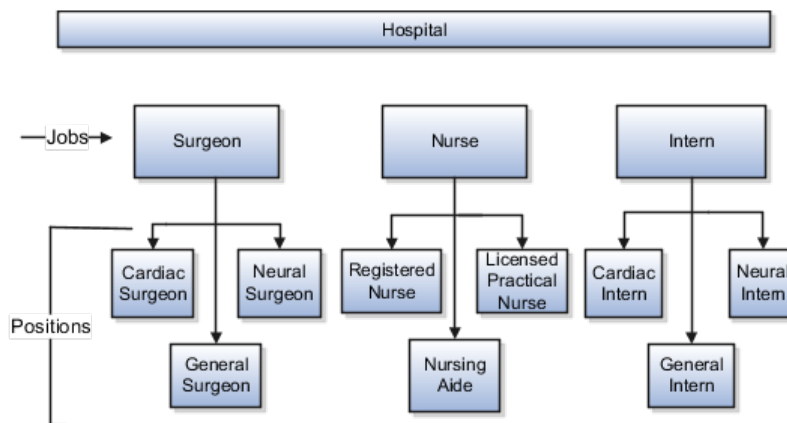


Health Care Industry

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

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

This figure illustrates the hospital position setup.



Jobs: Example

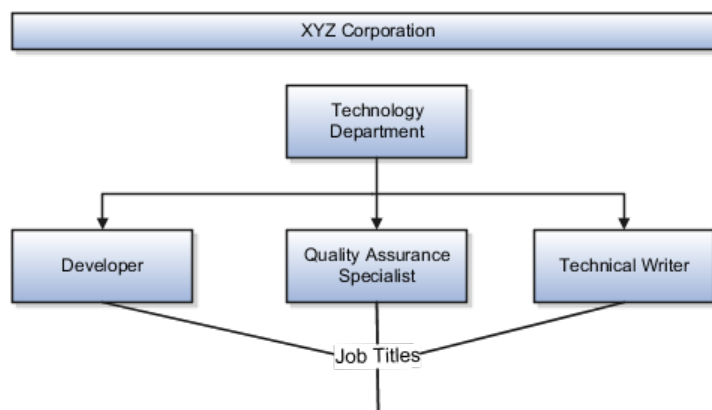
Jobs are typically used without positions by service industries where flexibility and organizational change are key features.

Software Industry

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

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

This figure illustrates the software industry job setup.



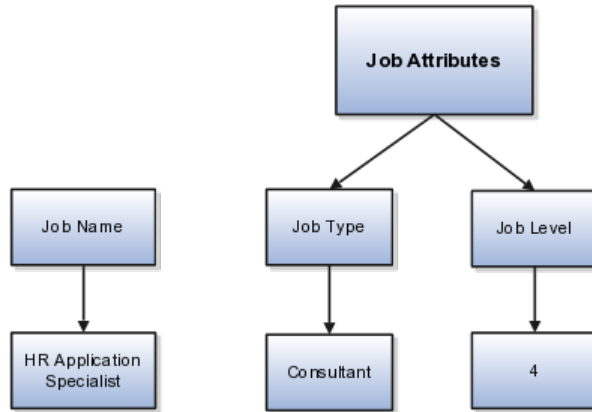
Job and Position Structures: Explained

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

Enterprise-Level Job Attributes

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

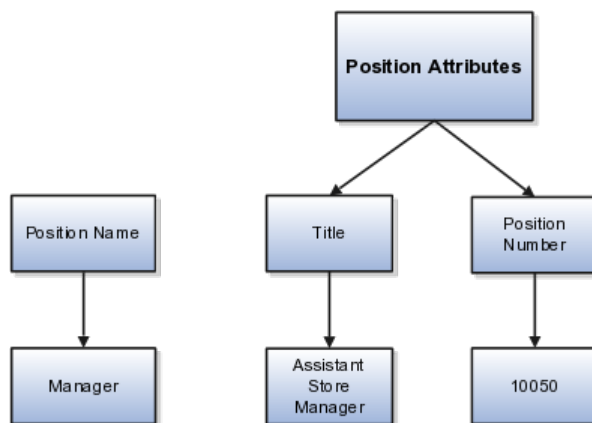
This figure illustrates how job type and job level provide further details for the HR Application Specialist job.



Enterprise-Level Position Attributes

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

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



Business Unit-Level Attributes for Positions

If you have information that you want to capture for positions that is specific to each business unit, then you can define attributes at the business unit level for positions. When you create positions, these attributes appear in addition to any enterprise-level attributes. For example, you may want to identify the sales

region for all positions in the sales business unit. You can set up a text attribute called Sales Region and use it to enter the necessary information when creating positions for the sales business unit.

Reference Data Set-Level Attributes for Jobs

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

FAQs for Define Initial Configuration

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

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

What's an ultimate holding company?

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

What's the default reference data set?

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

What happens if I override the set assignment?

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

Define Reference Data Sharing

Reference Data Sharing: Explained

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

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

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

Note

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

Reference Data Sets: Explained

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

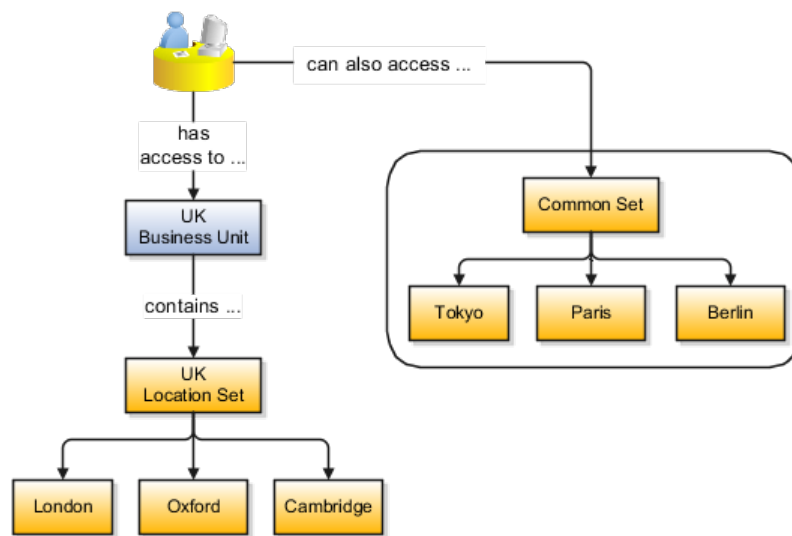
Consider the following scenario.

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

Partitioning

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

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



Reference Data Sets and Sharing Methods: Explained

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

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

Reference Data Sets

You begin this part of your implementation by creating and assigning reference data to sets. Make changes carefully as changes to a particular set will affect all business units or application components using that set. You can assign a

separate set to each business unit for the type of object that is being shared. For example, assign separate sets for payment terms, transaction types, and sales methods to your business units.

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

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

Reference Data Sharing Methods

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

- Assignment to one set only, no common values allowed. The simplest form of sharing reference data that allows assigning a reference data object instance to one and only one set. For example, Asset Prorate Conventions are defined and assigned to only one reference data set. This set can be shared across multiple asset books, but all the values are contained only in this one set.
- Assignment to one set only, with common values. The most commonly used method of sharing reference data that allows defining reference data object instance across all sets. For example, Receivables Transaction Types are assigned to a common set that is available to all the business units without the need to be explicitly assigned the transaction types to each business unit. In addition, you can assign a business unit specific set of transaction types. At transaction entry, the list of values for transaction types includes transaction types from the set assigned to the business unit, as well as transaction types assigned to the common set that is shared across all business units.
- Assignment to multiple sets, no common values allowed. The method of sharing reference data that allows a reference data object instance to be assigned to multiple sets. For instance, Payables Payment Terms use this method. It means that each payment term can be assigned to one or more than one set. For example, you assign the payment term Net 30 to several sets, but the payment term Net 15 is assigned to only your corporate business unit specific set. At transaction entry, the list of values for payment terms consists of only one set of data; the set that is assigned to the transaction's business unit.

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

Assigning Reference Data Sets to Reference Objects: Points to Consider

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

Determinant Types

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

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

Determinant

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

Reference Groups

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

Note

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

Items and Supplier Site Reference Data Sharing: Explained

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

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

Items

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

Suppliers Sites

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

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

- Supplier qualification data, such as approved supplier lists
- Catalog content, such as agreements, smart forms, public shopping lists, and content zones
- Procurement configuration data

Reference Data Sharing in Order Orchestration: Explained

Order Orchestration uses reference data sharing in the following areas: Manage Orchestration Process Definitions and Manage Hold Definitions. A set is a collection of business units.

Sets in Orchestration Process Definitions

When you create an orchestration process definition you must assign it to a set. During decomposition when an orchestration process is assigned to an order, the business unit of the order is checked to ensure that it belongs to the set for which the orchestration process definition was created.

Sets in Hold Definitions

The Set column of the Manage Hold Definitions page has a list of sets that can be assigned to hold codes. This means that hold codes can be specific to business units or, sets of business units can share hold codes.

FAQs for Define Reference Data Sharing

What reference data objects can be shared across business units?

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

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

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

What reference data objects can be shared across asset books?

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

Application Name	Reference Data Object	Method of Sharing
Assets	Bonus Rules	Assignment to one set only, no common values allowed

Assets	Depreciation Ceilings	Assignment to one set only, no common values allowed
Assets	Depreciation Methods	Assignment to one set only, with common values
Assets	Asset Descriptions	Assignment to one set only, no common values allowed
Assets	Property Types	Assignment to one set only, with common values
Assets	Prorate Conventions	Assignment to one set only, no common values allowed
Assets	Asset Queue Names	Assignment to one set only, with common values
Assets	Retirement Types	Assignment to one set only, with common values
Assets	Unplanned Types	Assignment to one set only, with common values

What reference data objects can be shared across cost organizations?

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

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

What reference data objects can be shared across project units?

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

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

Define Geographies

Defining Address Cleansing: Explained

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

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

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

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

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

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

Geography Structure

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

Geography Hierarchy

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

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

Geography Validation

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

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

Geography Structures: Explained

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

You can use the geography structure to establish:

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

How Geographies Can Be Related

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

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

Note

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

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

The Types of Geographies You Can Define for the Country

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

Note

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

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

Geography Hierarchy: Explained

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

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

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

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

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

Geography

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

Geography Type

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

Geography Usage

Geography usage indicates how a geography type or geography is used in the application. A master reference geography always has the usage of Master

Reference. User defined zones can have the usages of Tax, Shipping, or Territory, based on what is relevant for their purpose.

Master Reference Geography Hierarchy

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

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

User Defined Zones

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

Geography Validation: Explained

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

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

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

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

Map to Attribute

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

address attributes that appear are based on address formats delivered with the application, or your customer defined address formats.

Note

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

Enable List of Values

Once a geography type is mapped to an attribute, then you can specify whether the geography type will appear in a list of values during address entry in user interfaces. It is very important to review carefully if you want to enable a list of values. You should only enable a list of values if you have sufficient geography data imported or created for that geography. Once you have enabled a list of values for an address attribute, you can only select the geography data available for the geography type. This means that if a specific geography value is not available in the geography hierarchy, you cannot create an address with a different geography value.

Tax Validation

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

Geography Validation

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

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

Note

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

Geography Validation Control

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

These are the geography validation levels you can choose:

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

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

Note

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

Importing Geographies: Explained

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

to import the data. Having a good understanding of the import entity, interface table, and destination table will help you prepare your import data.

Consider the following when importing geographies:

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

File-Based Import Option

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

Geography Loader Process Option

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

Import Object Entity, Interface Table, and Destination Tables

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

Note

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

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

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

Importing 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 Fusion Applications 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 Fusion Applications represent the same data?
- Do you have to configure values in Oracle Fusion Applications to map to your data values?
- Do you have to customize Oracle Fusion Applications 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 Fusion Applications in order to be able to map your legacy data to the data needed by Oracle Fusion Applications. First, you must understand how Oracle Fusion Applications represent 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 Fusion Applications incorporate 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 Fusion Applications 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 Fusion Applications 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 Fusion Applications attributes. The validation information includes the navigation to the task where you can define values in Oracle Fusion Applications. For example, if you have values in your data that correlate to a choice list in Oracle Fusion Applications, 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.

Import Object	Related Import Object Topic
Country Structure	Country Structure Import Objects: How They Work Together

Extensible Attributes

If you need to extend the Oracle Fusion Applications object to import your legacy or source data, you must use Oracle Fusion CRM 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 Fusion Applications, 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 Fusion Applications 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 Oracle Fusion CRM 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 for Oracle Fusion Applications (<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.

Target Import Object	Description	Reference Guide File Names
ImpGeoStructureLevel	Contains information that specifies a country's geography structure. Sample attributes: GeographyType, GeographyTypeName, LevelNumber, and ParentGeographyType. Reference attribute: CountryCode	HZ_IMP_GEO_STRUCTURE_LEVELS_Reference

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 Fusion Applications represent the same data?
- Do you have to configure values in Oracle Fusion Applications 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 Fusion Applications in order to be able to map your legacy data to the data needed by Oracle Fusion Applications. First, you must understand how Oracle Fusion Applications represent 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 Fusion Applications incorporate 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 Fusion Applications 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 Fusion Applications 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 Fusion Applications attributes. The validation information includes the navigation to the task where you can define values in Oracle Fusion Applications. For example, if you have values in your data that correlate to a choice list in Oracle Fusion Applications, 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.

Import Object	Related Import Object Topic
ImpGeography	Geography Import Objects: How They Work Together

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

Extensible Attributes

Oracle Fusion Applications do not support extensible attributes for geographies. You can only import data for attributes provided by Oracle Fusion Applications.

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 Fusion Applications provide 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 Fusion Applications, 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 Fusion Applications 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 for Oracle Fusion Applications (<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 Geography File-Based Data Import asset.

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

Importing Geographies Using File-based Data Import: Worked Example

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

The following table summarizes the key decisions for this scenario:

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

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

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

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

Prerequisites when importing additional geography data after your initial import

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

Determine what information is in the source file

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

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

Create and schedule the import activity

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

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

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

Note

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

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

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

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

Note

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

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

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

8. Click **Next**.

Monitor the import results

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

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

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

Importing and Exporting Territory Geography Zones: Explained

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

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

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

4. Import the territory geography zones into another Oracle Fusion applications instance.

Import the master reference geography data

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

Define your territory geography zones

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

Export the territory geography zones

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

Import the territory geography zones

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

Note

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

Managing Geography Structures, Hierarchies, and Validation: Worked Example

This example shows how to 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.

Decisions to Consider	In This Example
Copy an existing country structure?	No, create a new country structure.
What is the structure of the geography types?	Create geography types with the following ranking structure: <ul style="list-style-type: none"> 1. County 2. Post Town
What is the geography hierarchy?	Create the following hierarchy: <ul style="list-style-type: none"> 1. Country of United Kingdom 2. County of Berkshire 3. Post Town of Reading
Which address style format will you use when mapping geography validations?	The default address style format, called the No Styles Format.
Are you using Oracle Fusion Tax for tax purposes?	No, do not select Tax Validation for the geography types.

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 **Create** button next to the **Copy Country Structure From** field.
4. In the Geography Structure section, select the County list item in the **Add Geography Type** field.
5. Click **Add**.
6. Select the Post Town list item in the **Add Geography Type** field.
7. Click **Add**.

Defining the geography hierarchy

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 Enterprise for Order Orchestration: Manage Enterprise HCM Information

Enterprise: Explained

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

Enterprise Defined

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

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

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

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

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

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

Define Enterprise for Order Orchestration: Manage Locations

Locations: Explained

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

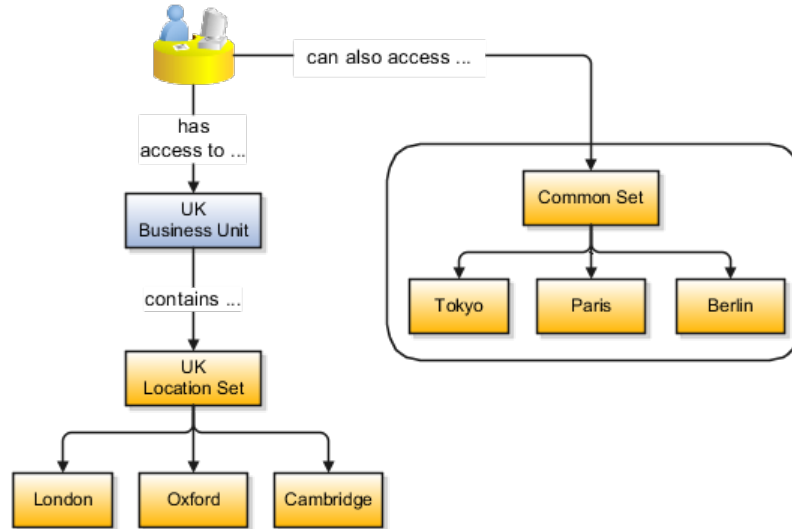
The locations that you create exist as separate structures that you can use for reporting purposes, and also in rules that determine employee eligibility for various types of compensation and benefits. You enter information about a location only once. Subsequently, when you set up other workforce structures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

Location Sets

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

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

The following figure shows how locations sets restrict access to users.



Creating Multiple Locations Simultaneously

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

FAQs for Define Enterprise for Order Orchestration: 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 Business Units: Manage Business Units

Business Units: Explained

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

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

The Oracle Fusion Applications business unit model:

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

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

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

In summary, use business units in the following ways:

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

Brief Overview of Business Unit Security

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

For example, if a payables invoicing business function is enabled, then it is clear that there are employees in this business unit that perform the function of payables invoicing, and need access to the payables invoicing functionality. Therefore, based on the correspondence between the business function and the job roles, appropriate data roles are generated automatically. Use Human Capital Management (HCM) security profiles to administer security for employees in business units.

Define Facilities for Order Orchestration

Item Organization: Explained

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

Note

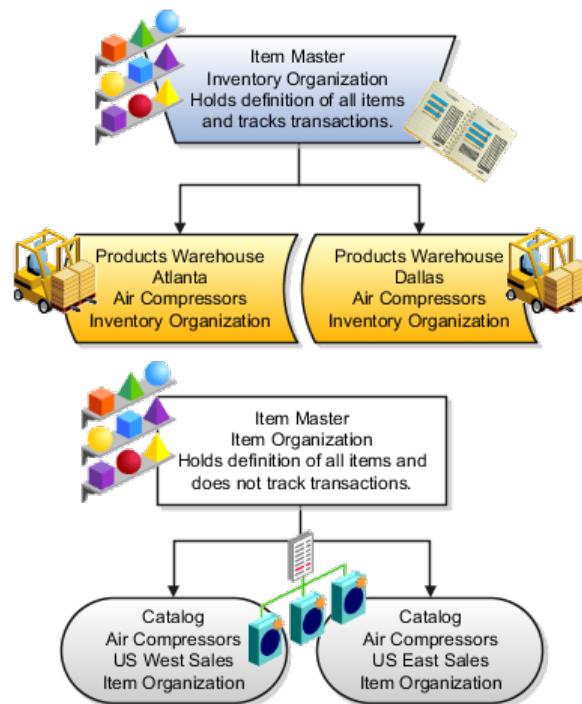
- Items belong to an item organization.
- Item attributes that are associated with financial and accounting information are hidden from the item if it exists within the item organization.

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

Item Master Organization: Explained

An item master organization lists and describes items that are shared across several inventory organizations or item organization.

The following example shows the choice between inventory organizations that track inventory transactions, stored in two warehouses, and item organizations that just track items, listed in two sales catalogs.



For the most efficient processing, you should:

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

Note

Oracle Fusion allows multiple item masters, however, use this capability cautiously. If you acquire a company, there may be value in allowing the old

item master to exist for a transition period. If you manage your subsidiaries as separate businesses, there may be reduced value in a single item master.

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 (Oracle Fusion Applications Edition).
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

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

See: Managing Oracle Fusion Applications Data Security Policies

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

See: Common Applications Configuration: Define Security for Human Capital Management

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

See: Oracle Fusion Applications Common Security Reference Manual

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

See: Getting Started With Oracle Authorization Policy Manager

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

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

- Oracle Fusion Applications is certified to integrate with Applications Access Controls Governor (AACG) in the Oracle 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 Customer Relationship Management (CRM) Self Service, respectively, and associate them with users.

Before Importing Users

Oracle Identity Management (OIM) handles importing users.

If legacy employees, contingent workers, and their assignments exist, the HCM Application Administrator imports these definitions by performing the 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 CRM.

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

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

Import Users

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

If no legacy users or user accounts can be imported or exist in an LDAP repository accessible to Oracle Identity Management (OIM), the IT security manager creates users manually in OIM or uses the 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, CRM or Oracle Fusion Supplier Portal.

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

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

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

Manage Duties

A person with a job role must be able to perform certain duties. In the Oracle Fusion Applications security reference implementation, enterprise roles inherit

duties through a role hierarchy. Each duty corresponds to a duty role. Duty roles specify the duties performed within applications and define the function and data access granted to the enterprise roles that inherit the duty roles.

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

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

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

Tip

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

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

Note

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

Manage Application Access Controls

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

Note

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

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

Security Tasks and Oracle Fusion Applications: How They Fit Together

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

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

- Security setup (Define Common Applications Configuration activity)
 - Define Implementation Users task group (optional)
 - Create Implementation Users task
 - Create Data Role for Implementation Users task
 - Provision Roles to Implementation Users task
 - Define security - tasks vary depending on deployed Oracle Fusion product family
 - Revoke Data Role from Implementation Users task
 - Import Worker Users task
 - Import Partner Users task
 - Manage Duties task
 - Manage Job Roles task
 - Manage Application Access Controls task
 - Define Automated Governance, Risk, and Performance Controls activity
 - Manage Application Access Controls task (AACG settings)
 - Manage Application Preventive Controls task
 - Manage Application Transaction Controls task
 - Manage Application Configuration Controls task
- User and role provisioning tasks
 - Implement Role Request and Provisioning Controls activity
 - Import Worker Users task
 - Import Partner Users task
 - Self Request User Roles task
 - Approve User and Role Provisioning Requests task

- Assign User Roles task
- Manage Supplier User Roles and User Role Usages task
- Map and Synchronize User Account Details task
- Tasks for viewing account details for self or others
- Tasks for applying and managing various role provisioning rules
- Tasks for running synchronization processes
- Security implementation and ongoing maintenance after setup (Manage IT Security activity)
 - Implement Function Security Controls
 - Create Job Role task
 - Import Worker Users task
 - Import Partner Users task
 - Manage Duties task
 - Manage Job Roles task
 - Manage Users task
 - Implement Data Security Controls
 - Manage Data Security Policies task
 - Manage Role Templates task
 - Manage Encryption Keys task
 - Manage Segment Security task
 - Manage Data Access Sets task
 - Define Security Profiles task group
 - Auditing tasks
 - Manage Security Audit, Compliance and Reporting activity
 - Manage Application Access Controls task

Note

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

Required Roles

The 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.

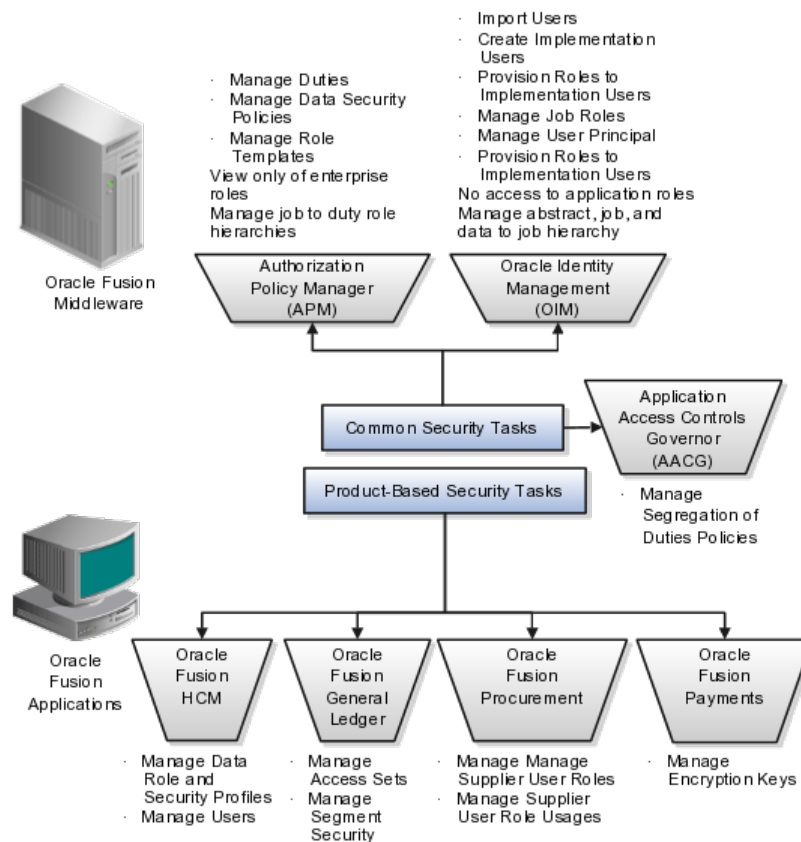
Important

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.

Example Task	Oracle BPM Task	Supporting Tools	Details
View duty roles inherited by a job role	Manage Duties	<ul style="list-style-type: none"> Authorization Policy Manager (APM) 	Each logical partition or pillar contains a collection of application roles, and function and data security policies.
View entitlement or policies carried by a job role	Manage Duties	<ul style="list-style-type: none"> APM 	In LDAP, the policy store stores application roles and the identity store stores enterprise roles.
Add a job role to a role hierarchy	Manage Job Roles	<ul style="list-style-type: none"> Oracle Identity Management (OIM) 	The identity store in LDAP stores enterprise roles.
Add a duty role to a role hierarchy	Manage Duties	<ul style="list-style-type: none"> APM 	LDAP stores the role hierarchy and the spanning of roles across multiple pillars or logical partitions.
Create a hierarchy of enterprise (abstract, job, data) roles	Manage Job Roles	<ul style="list-style-type: none"> OIM 	
Create a hierarchy of (application) duty roles	Manage Duties	<ul style="list-style-type: none"> APM 	
Create a new job role	Manage Job Roles	<ul style="list-style-type: none"> OIM 	The identity store in LDAP stores enterprise roles.
Change duty roles inherited by a job or abstract role	Manage Duties	<ul style="list-style-type: none"> APM 	The policy store stores duty roles. The identity store stores enterprise roles. Some duty roles may enable actions and their associated users interface features that your enterprise does not want users to perform in Oracle Fusion applications.
Create a new duty role	Manage Duties	<ul style="list-style-type: none"> APM 	All functions and actions in Oracle Fusion Applications that need to be secured are covered by the reference implementation. In some cases, especially with function customizations, a new duty role may be needed.

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

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

Provision roles to a user	<ol style="list-style-type: none"> 1. Provision Roles to Implementation Users 2. Manage Users 	<ul style="list-style-type: none"> • OIM • Oracle Fusion HCM • Oracle Fusion CRM • Oracle Fusion Suppliers 	<p>Implementation users are provisioned through OIM since HCM is not setup at the start of the implementation. The Provision Roles to Implementation Users is not needed once implementation is complete.</p> <p>Once HCM is setup, HCM is used to provision roles to non-implementation users by performing the Manage Users task. Human Resources (HR) transaction flows such as Hire and Promote also provision roles.</p> <p>Once supplier users are setup, Supplier Model can be used by internal users to maintain supplier user accounts or supplier users can maintain their accounts in Supplier Portal.</p>
View the job, abstract, and data roles provisioned to a user	<ol style="list-style-type: none"> 1. Manage Users 2. Manage User Principal 3. Provision Roles to Implementation Users 	<ul style="list-style-type: none"> • Human Capital Management (HCM) • OIM 	<p>LDAP stores users, roles and provisioning information.</p> <p>The Manage User Principal and Provision Roles to Implementation Users tasks are not needed once implementation is complete.</p>
Revoke role from user.	Manage Users	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • OIM 	
View audit logs	Not applicable	<ul style="list-style-type: none"> • 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 for Supply Chain Management

Data Security: Explained

By default, users are denied access to all data.

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

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

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

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

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

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

Data is secured by the following means.

Data security feature	Does what?
Data security policy	Grants access to roles by means of entitlement
Role	Applies data security policies with conditions to users through role provisioning.
Data role template	Defines the data roles generated based on enterprise setup of data dimensions such as business unit.

HCM security profile	Defines data security conditions on instances of object types such as person records, positions, and document types without requiring users to enter SQL code
Masking	Hides private data on non-production database instances
Encryption	Scrambles data to prevent users without decryption authorization from reading secured data

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

Data Security Policies

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

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

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

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

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

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

Note

Some data security policies are not defined as grants but directly in applications code. The security reference manuals for Oracle Fusion Applications offerings differentiate between data security policies that define a grant and data security policies defined in Oracle Fusion applications code.

A business object participating in a data security policy is the database resource of the policy.

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

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

Important

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

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

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

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

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

Data Roles

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

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

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

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

A data role may be granted entitlement over a set people.

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

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

HCM Security Profiles

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

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

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

Masking and Encryption

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

Defining Data Security After Enterprise Setup: Points to Consider

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

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

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

These tasks address data security administration. For information on using the user interface pages for setting up and managing data security, see the Oracle Fusion Middleware Administrator's Guide for Authorization Policy Manager (Oracle Fusion Applications edition).

Note

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

Manage Data Access Sets

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

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

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

Manage Segment Security

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

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

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

Manage Role Templates

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

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

Manage Data Security Policies

Data security grants provisioned to roles are data security policies. The security reference implementation provides a comprehensive set of predefined data

security policies and predetermined data security policies based on data role templates.

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

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

Warning

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

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

Manage Encryption Keys

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

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

Data Security in the Security Reference Implementation: Explained

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

The security implementation of an enterprise is likely a subset of the reference implementation, with the enterprise specifics of duty roles, data security policies, and HCM security profiles provided by the enterprise.

The business objects registered as secure in the reference implementation are database tables and views.

Granting or revoking object entitlement to a particular user or group of users on an object instance or set of instances extends the base Oracle Fusion Applications

security reference implementation without requiring customization of the applications that access the data.

Data Security Policies in the Security Reference Implementation

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

Tip

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

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

Warning

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

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

HCM Security Profiles in the Security Reference Implementation

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

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

Data Roles

The security reference implementation includes no predefined data roles to ensure a fully secured initial Oracle Fusion Applications environment.

The security reference implementation includes data role templates that you can use to generate a set of data roles with entitlement to perform predefined business functions within data dimensions such as business unit. Oracle Fusion Payables invoicing and expense management are examples of predefined business functions. Accounts Payable Manager - US is a data role you might

generate from a predefined data role template for payables invoicing if you set up a business unit called US.

HCM provides a mechanism for generating HCM related data roles.

Securing Data Access: Points to Consider

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

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

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

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

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

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

Grants

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

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

Grants control which data a user can access.

Note

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

A grant logically joins a user or role and an entitlement with a static or parameterized object instance set. For example, `REGION='WEST'` is a static object instance set and `REGION=&GRANT_ALIAS.PARAMETER1` is a parameterized object instance set. In the context of a specific object instance, grants specify the allowable actions on the set of accessible object instances. In the database, grants are stored in `FND_GRANTS` and object instance sets are stored in `FND_OBJECT_INSTANCE_SETS`. Object access can be tested using the privilege check application programming interface (API).

Securing a Business Object

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

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

Note

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

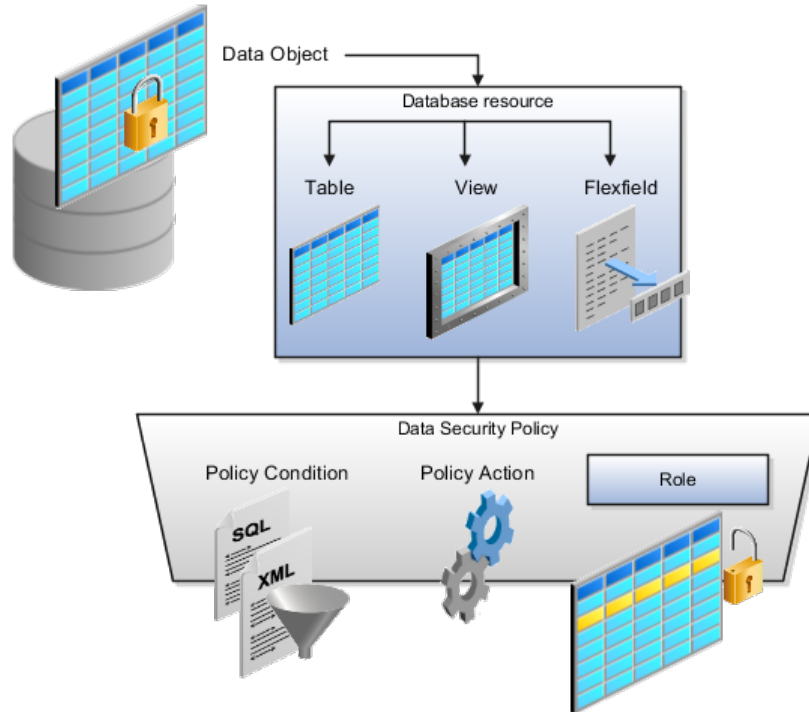
Manage Data Security Policies

Database Resources and Data Security Policies: How They Work Together

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

The database resource defines and instance of a data object. The data object is a table, view, or flexfield.

The following figure shows the database resource definition as the means by which a data security policy secures a data object. The database resource names the data object. The data security policy grants to a role access to that database resource based on the policy's action and condition.



Database Resources

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

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

Data Security Policies

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

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

Note

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

Actions

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

Conditions

A condition is either a SQL predicate or an XML filter. A condition expresses the values in the data object by a search operator or a relationship in a tree hierarchy. A SQL predicate, unlike an XML filter, is entered in a text field in the data security user interface pages and supports more complex filtering than an

XML filter, such as nesting of conditions or sub queries. An XML filter, unlike a SQL predicate, is assembled from choices in the UI pages as an AND statement.

Tip

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

Manage Role Templates

Data Role Templates: Explained

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

Note

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

The following attributes define a data role template.

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

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

Note

Abstract, job, and data roles are enterprise roles in Oracle Fusion Applications. Oracle Fusion Middleware products such as Oracle Identity Manager (OIM) and Authorization Policy Manager (APM) refer to enterprise roles as external roles. Duty roles are implemented as application roles in APM and scoped to individual Oracle Fusion Applications.

The dimension expresses stripes of data, such as territorial or geographic information you use to partition enterprise data. For example, business units are a type of dimension, and the values picked up for that dimension by the data role template as it creates data roles are the business units defined for your enterprise. The data role template constrains the generated data roles with grants of entitlement to access specific data resources with particular actions. The data role provides provisioned users with access to a dimensional subset of the data granted by a data security policy.

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

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

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

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

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

Making Changes To Data Role Templates

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

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

Making Changes to Dimension Values

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

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

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

Manage Data Role and Security Profiles

HCM Data Roles: Explained

HCM data roles, like all Oracle Fusion Applications data roles, define data security policies: they enable users to perform a set of tasks, using identified menus, menu items, and pages in application user interfaces, on a specified set of data within those user interfaces. Because data roles are specific to the enterprise, no predefined HCM data roles exist.

How HCM Data Roles Differ from Other Data Roles

HCM data roles differ from other data roles in the following ways:

- You create and maintain HCM data roles outside Oracle Identity Management (OIM) and the Oracle Fusion Middleware Authorization Policy Manager (APM), and they are not based on data role templates.

Although HCM data roles are visible in the Oracle Fusion Middleware APM, they must not be maintained there.

- A single HCM data role can enable access to data of multiple types.

You identify the data that users can access in HCM security profiles. You can create security profiles for the person, organization, position, country, legislative data group (LDG), document type, payroll, payroll flow, and workforce business process objects.

Selecting the Job Role

Each HCM data role is associated with a single job role, which you select from the list of enterprise roles. The HCM securing objects that the selected role needs to access are identified automatically, and the appropriate types of security profile are displayed. For example, if you select the job role human resource analyst, users with that job role need to access managed person, public person, organization, position, LDG, and document type data; therefore, security profiles for those object types must be included in the HCM data role. The security profile types that appear in the HCM data role vary according to the data requirements of the selected job role.

If you select a job role that requires no access to HCM data secured by security profiles, you cannot create an HCM data role.

Note

If you create custom job roles in OIM, you must add them to a locally defined role category that ends with "Job Roles"; otherwise, they do not appear in the list of job roles when you create an HCM data role. Do not add custom job roles to the predefined role category HCM - Job Roles.

Creating or Selecting the Security Profiles

You can either create new security profiles or use existing security profiles. For each object type, you can include only one security profile in an HCM data role.

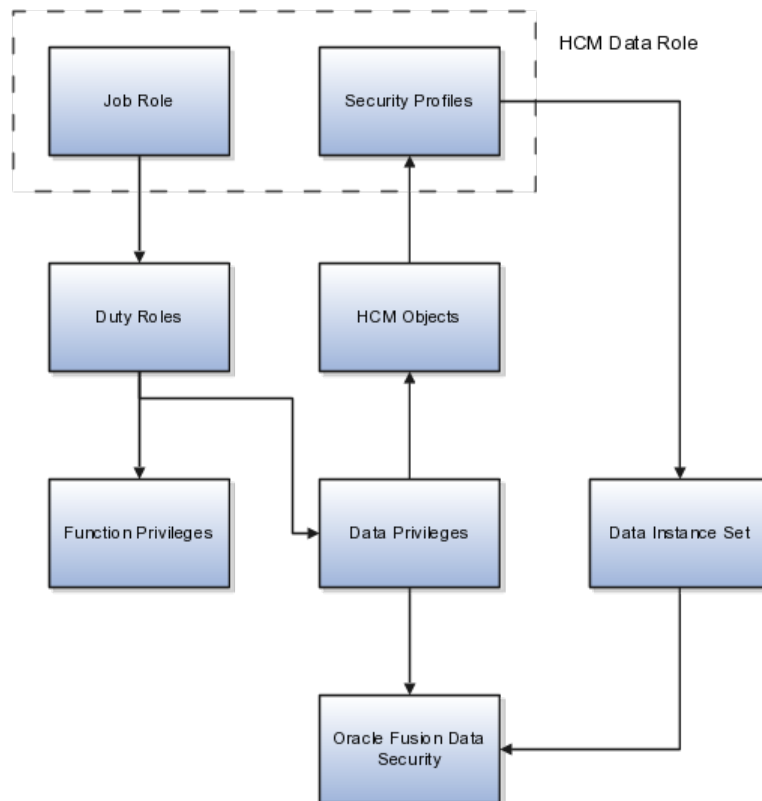
Users with Multiple HCM Data Roles

When users have multiple HCM data roles, the data security policies arising from each role remain separate. For example, being able to promote or terminate workers in the purchasing department in one HCM data role and view contact details of all workers in the sales department in another HCM data role does not enable a user to promote or terminate workers in the sales department.

Components of the HCM Data Role

The following figure summarizes how the components of the HCM data role contribute to Oracle Fusion Data Security for the data role. Oracle Fusion Data Security comprises the data security policies for data roles that are generated automatically when data roles are created.

The job role that you select in the HCM data role inherits multiple duty roles. Each duty role has one or more function privileges and related data privileges, from which the relevant HCM objects are identified. The specific instances of the objects required by this HCM data role are identified in security profiles and stored in a data instance set. Data security policy data is created automatically in Oracle Fusion Data Security when you create the data role.



For example, the human resource specialist job role inherits the employee hire and worker promotion duty roles, among many others. The inherited duty roles provide both function privileges, such as Hire Employee, Rehire Employee, and Promote Workers, and data privileges to HCM objects, such as person and assignment. The specific instances of those objects required by this HCM data role, such as people with assignments in a specified legal employer and department, are identified in security profiles.

HCM Security Profiles: Explained

A security profile defines the criteria that identify instances of a human capital management (HCM) object. For example, a person security profile defines the criteria that identify one or more person records, and a position security profile defines the criteria that identify one or more positions. When you include a security profile in an HCM data role and provision the data role to a user, that user can access the data instances identified in the security profile. The type of access available to the user (for example whether the user can edit or simply view the data) depends on the job role identified in the HCM data role.

HCM Object Types

You can create security profiles for the following HCM object types:

- Person
 - Managed person
 - Public person
- Organization
- Position
- Legislative data group (LDG)
- Country
- Document type
- Payroll
- Payroll flow
- Workforce business process

All security profile definitions for these HCM objects are eventually visible in the Oracle Fusion Middleware Authorization Policy Manager (APM). The name of the security profile's data instance set in the Oracle Fusion Middleware APM is derived from the name of the security profile and the relevant object type. For example, if the security profile name is Manager Hierarchy, then the data instance set for the object PER_ALL_PEOPLE_F is HCM:PER:PER_ALL_PEOPLE_F:Manager Hierarchy.

You must use the Oracle Fusion Human Capital Management interfaces, which are designed for ease of use and access, to create and maintain security profiles; do not use the Oracle Fusion Middleware APM to maintain security profiles for these HCM objects.

Security Criteria in HCM Security Profiles

In any HCM security profile, you specify the criteria that identify data instances of the relevant type. For example, in an organization security profile, you can identify organizations by organization hierarchy, by organization classification, or by listing organizations to include in or exclude from the security profile. All of the criteria in an HCM security profile apply when the data instance set is defined; for example, if you identify organizations by both organization

hierarchy and organization classification, then both sets of criteria apply, and only those organizations that satisfy all criteria belong to the data instance set.

Predefined HCM Security Profiles

The following HCM security profiles are predefined:

Security Profile Name	HCM Security Profile Type	Description
View All People	Person	Identifies all person records in the enterprise
View Own Record	Person	Identifies the signed-on user's own person record and the person records of that user's contacts
View Manager Hierarchy	Person	Identifies the signed-on user's line manager hierarchy
View All Workers	Person	Identifies the person records of all people who have a work relationship in the enterprise
View All Organizations	Organization	Identifies all organizations in the enterprise
View All Positions	Position	Identifies all positions in the enterprise
View All Legislative Data Groups	LDG	Identifies all LDGs in the enterprise
View All Countries	Country	Identifies all countries in the FND_TERRITORIES table
View All Document Types	Document Type	Identifies all document types in the enterprise
View All Payrolls	Payroll	Identifies all payrolls in the enterprise
View All Flows	Payroll Flow	Identifies all payroll flows in the enterprise
View All Workforce Business Processes	Workforce Business Process	Identifies all registered workforce business processes in the enterprise

You can include the predefined security profiles in any HCM data role, but you cannot edit them. Note also that the **View all** option is disabled in any security profile that you create; this restriction exists because predefined security profiles exist for this requirement.

Creating Security Profiles

You can create security profiles either individually or as part of the process of creating an HCM data role. If you have standard requirements, it may be more efficient to create the security profiles individually and include them in appropriate HCM data roles.

Reusability and Inheritance of Security Profiles

Regardless of how you create them, all security profiles are reusable; they do not belong to particular HCM data roles, and you can include them in any HCM data role for which they define an appropriate data instance set.

You can include security profiles in other security profiles. For example, you can include an organization security profile:

- In a person security profile, to secure person records by department, business unit, or legal employer
- In a position security profile, to secure positions by department or business unit

Therefore, one security profile can inherit the data instance set defined by another.

Assigning Security Profiles to Abstract Roles: Explained

Abstract roles define a worker's role in the enterprise independently of the job that the worker is hired to do.

These abstract roles are predefined in Oracle Fusion Human Capital Management:

- Line manager
- Employee
- Contingent worker

Enabling Data Access for Abstract Roles

Typically, you create role mappings during implementation to provision abstract roles automatically to eligible workers. Although users with these roles may be able to sign in to Oracle Fusion Applications and navigate to tasks of interest, they have no automatic access to data. For example, employees can navigate to the Person Gallery but cannot view portraits or see lists of person names in product interfaces, and line managers can navigate to the Manager Resources Dashboard but can see no data for their organizations. To enable users with abstract roles to access relevant HCM data, you must assign security profiles to those abstract roles.

Predefined Security Profiles to Assign to Abstract Roles

To enable users with abstract roles to access relevant data, you assign the following predefined security profiles directly to the employee, contingent worker, and line manager abstract roles.

Security Profile Type	Employee	Contingent Worker	Line Manager
Person	View Own Record	View Own Record	View Manager Hierarchy
Public person	View All Workers	View All Workers	View All Workers
Organization	View All Organizations	View All Organizations	View All Organizations
Position	View All Positions	View All Positions	View All Positions
Legislative data group	View All Legislative Data Groups	View All Legislative Data Groups	View All Legislative Data Groups
Country	View All Countries	View All Countries	View All Countries

Document type	View All Document Types	View All Document Types	View All Document Types
Payroll Flow	Not applicable	Not applicable	View All Flows
Workforce Business Process	Not applicable	Not applicable	View All Workforce Business Processes

After implementation, you may want to change aspects of this data access. For example, you may want to create your own security profiles and assign those directly to abstract roles; however, you must remember that such changes apply to all users who have the abstract role.

HCM Data Roles

Users who have abstract roles are likely to gain additional data access by means of HCM data roles that you define for their job roles. For example, you may create an HCM data role for human resource specialists to enable them to access the person records of all workers in a legal employer. Such data access is in addition to any data access provided by abstract roles.

Assigning Security Profiles to Abstract Roles: Worked Example

This example shows how to assign predefined security profiles to the employee, contingent worker, and line manager abstract roles.

Searching for the Employee Abstract Role

1. On the All Tasks tab of the Overview page of the Setup and Maintenance work area, search for the task Manage Data Role and Security Profiles.
2. In the Search Results region, click **Go to Task**.
3. On the Manage Data Roles and Security Profiles page, enter the abstract-role name Employee in the **Role** field. Click **Search**.
4. In the Search Results region, highlight the entry for the predefined Employee role and click **Assign**.

Assigning Security Profiles to the Employee Abstract Role

1. On the Assign Data Role: Security Criteria page, select the security-profile values shown in the following table. These are the security profiles that are typically assigned to the employee role. You may see a subset of these security profiles, depending on the combination of product offerings that you are implementing.

Field	Value
Organization Security Profile	View All Organizations
Position Security Profile	View All Positions
Country Security Profile	View All Countries
LDG Security Profile	View All Legislative Data Groups

Person Security Profile (Person section)	View Own Record
Person Security Profile (Public Person section)	View All Workers
Document Type Security Profile	View All Document Types

2. Click **Review**.
3. On the Assign Data Role: Review page, click **Submit**.
4. On the Manage Data Roles and Security Profiles page, search again for the predefined Employee role.
5. In the Search Results region, confirm that a green check mark appears in the **Security Profiles** column for the Employee role. The check mark confirms that security profiles are assigned to the role.

Repeat the steps in Searching for the Employee Abstract Role and Assigning Security Profiles to the Employee Abstract Role for the predefined Contingent Worker role.

Searching for the Line Manager Abstract Role

1. On the Manage Data Roles and Security Profiles page, enter the abstract-role name Line Manager in the **Role** field. Click **Search**.
2. In the Search Results region, highlight the entry for the predefined Line Manager role and click **Assign**.

Assigning Security Profiles to the Line Manager Abstract Role

1. On the Assign Data Role: Security Criteria page, select the security-profile values shown in the following table. These are the security profiles that are typically assigned to the line manager role. You may see a subset of these security profiles, depending on the combination of product offerings that you are implementing.

Field	Value
Organization Security Profile	View All Organizations
Position Security Profile	View All Positions
LDG Security Profile	View All Legislative Data Groups
Person Security Profile (Person section)	View Manager Hierarchy
Person Security Profile (Public Person section)	View All Workers
Document Type Security Profile	View All Document Types
Payroll Flow	View All Flows
Workforce Business Process	View All Workforce Business Processes

2. Click **Review**.
3. On the Assign Data Role: Review page, click **Submit**

4. On the Manage Data Roles and Security Profiles page, search again for the predefined Line Manager role.
5. In the search results, confirm that a green check mark appears in the **Security Profiles** column for the Line Manager role. The check mark confirms that security profiles are assigned to the role.

Define Users for Supply Chain Management

Securing Identities and Users: Points To Consider

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

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

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

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

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

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

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

Stores

Various locations store identity and user data.

Identity data consists of the following.

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

In Oracle Fusion Applications, identities and users correspond one to one, but not all identities correspond to a user, and not all users are provisioned with an identity. Some identities stored in HR and Trading Community Model may not be provisioned to user accounts and therefore are not synchronized with Oracle

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

Importing Users

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

Your tasks may include the following.

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

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

Provisioning Events

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

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

Deactivating or terminating an identity triggers revocation of some roles to end all assignments, but may provision new roles needed for activities, such as

a pay stub review. If the corresponding user for the identity was provisioned with a buyer role, terminating the identity causes the user's buyer record in Procurement to be disabled, just as the record was created when the user was first provisioned with the buyer role.

Notifications and Audits

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

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

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

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

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

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

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

Delegated Administration

You can designate local administrators as delegated administrators to manage a subset of users and roles.

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

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

You can also define delegated administration policies based on roles. You authorize users provisioned with specific roles named in the policy to request a

subset of roles for themselves if needed, such as authorizing a subset of roles for a subset of people. For example, the policy permits a manager of an Accounts Payables department to approve a check run administrator role for one of their subordinates, but prohibits the delegated administrator from provisioning a budget approver role to the subordinate.

Credentials

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

Applications themselves must be credentialed to access one another.

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

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

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

Manage HCM Role Provisioning Rules

Role Provisioning and Deprovisioning: Explained

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

Role Provisioning Methods

Roles can be provisioned to users:

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

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

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

Role Types

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

Automatic Role Provisioning

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

Role Deprovisioning

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

Automatically provisioned roles can be deprovisioned manually at any time.

Manually provisioned roles are deprovisioned automatically only when all of the user's work relationships are terminated; in all other circumstances, users retain manually provisioned roles until they are deprovisioned manually.

Changes to Assignment Managers

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

Roles at Termination

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

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

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

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

Date-Effective Changes to Assignments

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

Role Mappings: Explained

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

The role mapping can support:

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

Automatic Provisioning of Roles to Users

A role is provisioned to a user automatically if:

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

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

Attribute	Value
Department	Finance Department
Job	Sales Manager

Assignment Status	Active
-------------------	--------

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

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

Note

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

Manual Provisioning of Roles to Users

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

Users can provision a role to other users if:

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

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

Attribute	Value
Manager with Reports	Yes
Assignment Status	Active

Any user with at least one assignment that satisfies both of these conditions can provision the role Quality Assurance Team Leader manually to other users, who are typically direct and indirect reports.

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

Role Requests from Users

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

Users can request a role if:

- At least one of their own assignments satisfies all conditions associated with the role mapping.
- You select the **Self-requestable** option for the role in the role mapping.

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

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

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

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

Immediate Provisioning of Roles

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

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

- Provisioned immediately to all users who do not currently have roles for which they are eligible
- Deprovisioned immediately from users who are no longer eligible for roles that they currently have

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

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

Role-Mapping Names

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

Name	Description
Autoprovisioned Roles Sales Department	Mapping includes all roles provisioned automatically to anyone in the sales department

Benefits Specialist Autoprovisioned	Mapping defines the conditions for autoprovisioning the Benefits Specialist role
Line Manager Requestable Roles	Mapping includes all roles that a line manager can provision manually to direct and indirect reports

Role Mappings: Examples

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

Creating a Role Mapping for Employees

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

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

Attribute	Value
System Person Type	Employee
Assignment Status	Active

In the role mapping you include the:

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

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

Note

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

Creating a Role Mapping for Line Managers

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

Attribute	Value
Business Unit	Sales

Assignment Status	Active
Manager with Reports	Yes

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

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

Creating a Role Mapping for Retirees

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

Attribute	Value
System Person Type	Retiree
Assignment Status	Inactive

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

Creating a Role Mapping for Sales Managers

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

Attribute	Value
Department	Sales
Job	Sales manager
Grade	6
Assignment Status	Active

In the role mapping, you include the:

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

Import Worker Users

Defining Security After Enterprise Setup: Points to Consider

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

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

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

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

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

Before Importing Users

Oracle Identity Management (OIM) handles importing users.

If legacy employees, contingent workers, and their assignments exist, the HCM Application Administrator imports these definitions by performing the 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 CRM.

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

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

Import Users

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

If no legacy users or user accounts can be imported or exist in an LDAP repository accessible to Oracle Identity Management (OIM), the IT security manager creates users manually in OIM or uses the 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, CRM or Oracle Fusion Supplier Portal.

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

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

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

Manage Duties

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

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

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

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

Tip

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

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

Note

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

Manage Application Access Controls

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

Note

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

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

Importing Worker Users: Explained

You can import workers from legacy applications to Oracle Fusion Applications using the Import Worker Users task. By enabling you to bulk-load existing data, this task is an efficient way of creating and enabling users of Oracle Fusion Applications.

The Import Worker Users Process

Importing worker users is a two-stage process:

1. On the Initiate Data Load page, you generate and complete the Create Worker spreadsheet. You must map your data to the spreadsheet columns and provide all required attributes. Once the spreadsheet is complete, you import the data to the HCM Data Loader stage tables.

HCM Data Loader is a generic utility for loading data to Oracle Fusion Human Capital Management from external sources.

2. In the Data Exchange work area, you run the Load Batch Data process to load data from the HCM Data Loader stage tables to the Oracle Fusion application tables.

User-Account Creation

Oracle Fusion user accounts are created automatically for imported workers in Oracle Identity Management (OIM), unless automatic account creation is disabled.

By default, user account names and passwords are sent automatically to users when their accounts are created. This default action may have been changed at enterprise level, as follows:

- User account names and passwords may be sent to an enterprise-wide e-mail rather than to users themselves.
- Automatic sending of user account names and passwords may be disabled for the enterprise; in this case, you can notify users at an appropriate time.

Role Provisioning

Once user accounts exist, roles are provisioned to users automatically in accordance with current role-provisioning rules. For example, current rules could provision the employee abstract role to every worker. Role provisioning occurs automatically unless it has been disabled for the enterprise.

Importing Worker Users: Worked Example

This example shows how to import worker users from legacy applications to Oracle Fusion Applications.

The following table summarizes key decisions for this task.

Decisions to Consider	In This Example
What are my spreadsheet names? You can define your own naming convention; in this example, the names are selected to make identifying the spreadsheet contents easy.	<ul style="list-style-type: none"> • WorkersMMDDYYBatchnn.xlsx • WorkersMMDDYYBatchnnErrorsnn.xlsx For example, Workers042713Batch01.xlsx.
What is my batch name?	Workers042713Batchnn
Where will I fix Load Batch Data errors?	In the spreadsheet Workers042713BatchnnErrorsnn.xlsx

Summary of the Tasks

Import worker users by:

1. Selecting the Import Worker Users task
2. Creating the spreadsheet
3. Entering workers in the spreadsheet

4. Importing the spreadsheet data to the HCM Data Loader stage tables
5. Loading workers to the application tables from the HCM Data Loader stage tables
6. Reviewing the results of the Load Batch Data process and correcting errors

Prerequisites

Before you can complete this task, you must have:

1. Installed the desktop client Oracle ADF 11g Desktop Integration
2. Enabled the Trust Center setting **Trust access to the VBA project object** in Microsoft Excel

Selecting the Import Worker Users Task

1. On the Overview page of the Setup and Maintenance work area, click the All Tasks tab.
2. In the Search region, complete the fields as shown in this table.

Field	Name
Search	Task
Name	Import Worker Users

3. Click **Search**.
4. In the search results, click **Go to Task** for the task Import Worker Users.

The task navigates to the Initiate Data Load page.

Alternatively, you can select the Import Worker Users task from an implementation project.

Creating the Spreadsheet

1. On the Initiate Data Load page, find the entry for Create Worker in the list of business objects.

Create Worker appears after other business objects such as departments, locations, and jobs, because those business objects (regardless of how you create them) must be created before worker users.

2. Click **Create Spreadsheet** for the Create Worker entry.
3. When prompted, save the spreadsheet locally using the name Workers042713Batch01.xlsx.
4. When prompted, sign in to Oracle Fusion Applications using your Oracle Fusion user name and password.

Entering Workers in the Spreadsheet

1. In the **Batch Name** field of the spreadsheet Workers042713Batch01.xlsx, replace the default batch name with the batch name Workers042713Batch01.

2. Enter workers in the spreadsheet.

Ensure that you provide any required values and follow instructions in the spreadsheet for creating additional rows.

Importing the Spreadsheet Data to the HCM Data Loader Stage Tables

Use the default values except where indicated.

1. In the workers spreadsheet, click **Upload**.
2. In the Upload Options window, click **OK**.

As each row of data is uploaded to the HCM Data Loader stage tables, its status is updated.

3. When uploading completes, identify any spreadsheet rows with the status **Insert Failed**, which indicates that the row failed to upload.
4. For any row that failed to upload, double-click the status value to display a description of the error.
5. When you have corrected any errors, click **Upload** again to load the remaining rows to the same batch.

Loading Workers to the Application Tables from the HCM Data Loader Stage Tables

Use the default values except where indicated.

1. In Oracle Fusion Applications, select **Navigator - Workforce Management - Data Exchange**.
2. In the Tasks pane of the Data Exchange work area, click **Load Batch Data**.
3. In the Search region of the Load Batch Data page, enter the batch name **Workers042713Batch01** in the **Batch Name** field.
4. Click **Search**.
5. In the Search Results region, select the batch **Workers042713Batch01** and click **Run**.
6. On the Schedule Request page, click **Submit**.

Reviewing the Results of the Load Batch Data Process and Correcting Errors

1. On the Load Batch Data page, search again for the batch **Workers042713Batch01** and review its status in the **Batch Status** column of the Search Results region.
2. If the batch status is **Complete**, click **Done** to close the Load Batch Data page; otherwise, continue with the next step.
3. If the batch completed with errors, select the batch in the search results.
4. On the Batch Summary tab of the Batch Details region, select **Action - Extract Failed Objects to Spreadsheet**.

An errors spreadsheet with a standard name is created automatically.

5. Save the errors spreadsheet locally as **Workers042713Batch01Errors01.xlsx**.

- Leave the batch name in the errors spreadsheet as Workers042713Batch01.
- Review the error messages in the spreadsheet and correct the errors.
 - Repeat this process from the task Importing the Spreadsheet Data to the HCM Data Loader Stage Tables until all errors are fixed.

If further errors occur, increment the errors-spreadsheet suffix by 1; for example, Workers042713Batch01Errors02, Workers042713Batch01Errors03, and so on.

- Click **Done** to close the Load Batch Data page.

To load a new batch of workers on the same date, increment the batch number in the spreadsheet and batch names; for example, Workers042713Batch02.

Manage Users

Creating Users: Worked Example

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

Note

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

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

Prerequisites

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

Attribute	Value
System Person Type	Employee

Assignment Status	Active
-------------------	--------

In the role mapping you include the:

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

Creating a User

1. On the Search Person page, click the **Create** icon to open the Create User page.
2. Complete the fields, as shown in this table:

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

3. In the User Details region, leave the User Name field blank. The user name defaults to the user's e-mail ID.
4. In the Employment Information region, select the person type **Employee** and complete the fields as shown in the table:

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

Assigning Roles to the User

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

User Details System Extract Report

The Oracle BI Publisher User Details System Extract Report includes details of some or all Oracle Fusion Applications user accounts.

To run this report, you must have an HCM data role that provides view-all access to person records for the Human Capital Management Application Administrator job role.

To run the report:

1. Navigate to **Tools - Reports and Analytics**.
2. In the Contents pane of the Reports and Analytics work area, navigate to **Shared Folders - Human Capital Management - Workforce Management - Human Resources Dashboard**.
3. Select the User Details System Extract report.
4. In the report window, click **More**.
5. On the Oracle Business Intelligence page for the report, select **Open** to run the report immediately or **Schedule** to schedule the report.

Parameters

User Population

Enter one of the following values to identify the group of user accounts to include in the report.

Value	Description
HCM	User accounts with an associated HCM person record.
TCA	User accounts with an associated TCA party account.
OIM	Accounts for users in the PER_USERS table who do not have an associated person number or party ID. OIM users are also referred to as implementation users.
ALL	HCM, TCA, and OIM users accounts.

From Date

Accounts for HCM and OIM users created on or after this date are included in the report. If you specify no **From Date** value, then accounts with any creation date are included, subject only to any **To Date** value that you specify.

From and to dates do not apply to the TCA user population; the report includes all TCA users if you include them in the report's user population.

To Date

Accounts for HCM and OIM users created on or before this date are included in the report. If you specify no **To Date** value, then accounts with any creation date are included, subject only to any **From Date** value that you specify.

From and to dates do not apply to the TCA user population; the report includes all TCA users if you include them in the report's user population.

User Active Status

Enter one of the following values to identify the user-account status.

Value	Description
A	Include active accounts, which belong to users with current roles.

I	Include inactive accounts, which belong to users with no current roles.
All	Include both active and inactive user accounts.

Report Results

The output is an XML-formatted file where user accounts are grouped by type, as follows:

- Group 1 (G_1) includes HCM user accounts.
- Group 2 (G_2) includes TCA party user accounts.
- Group 3 (G_3) includes OIM user accounts.

The information provided in the extract varies with the account type.

HCM User Accounts

Business Unit Name

The business unit from the primary work relationship.

Composite Last Update Date

The date when any one of a number of values, including assignment managers, location, job, and person type, was last updated.

Department

The department from the primary assignment.

Worker Type

The worker type from the user's primary work relationship.

Generation Qualifier

The user's name suffix (for example, Jr., Sr., or III).

Hire Date

The enterprise hire date.

Role Name

A list of roles currently provisioned to workers whose work relationships are all terminated. This value appears for active user accounts only.

Title

The job title from the user's primary assignment.

TCA User Accounts

Organizations

A resource group.

Roles

A list of job, abstract, and data roles provisioned to the user.

Managers

The manager of a resource group.

OIM User Accounts

Start Date

The date from when the account existed.

Created By

The user name of the user who created the account.

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.

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 predefined in Oracle Fusion Applications and available when creating SOD policies.

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

Enforcement

Oracle Fusion Applications enforces SOD policies under the following circumstances.

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

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

Note

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

A single SOD policy can include entitlement from multiple instances of a single enterprise resource planning environment. For example, one SOD policy is enforced in implementation, test, and production instances of Oracle Fusion Applications.

Managing Segregation of Duties Risks and Violations: Critical Choices

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

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

- Administrator of an external program such as the Procurement Administrator for the supplier portal or the Partner Manager for the PRM Program
- Senior executive spanning multiple organizations in an enterprise with opposing interests
- Risk management professional implementing an Oracle 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 Manger role entitles the user to enter orders, which could result in exploitation when filling shipping quotas. You can remove the entitlement to enter orders that the Order Manger job role inherits from the Orchestration Order Scheduling Duty role. Or you could segregate the shipping and order entry duties by defining an SOD policy that allows a user to have either job role but not both.

False Positives

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

You can reduce false positives by the following methods.

- Define exclusion conditions that can be applied to individual or groups of policies.

- Define logically complex SOD policies that enforce more exacting specifications.
- Determine whether conflicts should be prevented, monitored, or subjected to approval during provisioning.

Path Level Detection

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

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

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

Waiving or Accepting Violations

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

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

Resolving Conflicts

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

The following approaches resolve conflicts.

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

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

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

Role Provisioning and Segregation of Duties: How They Work Together

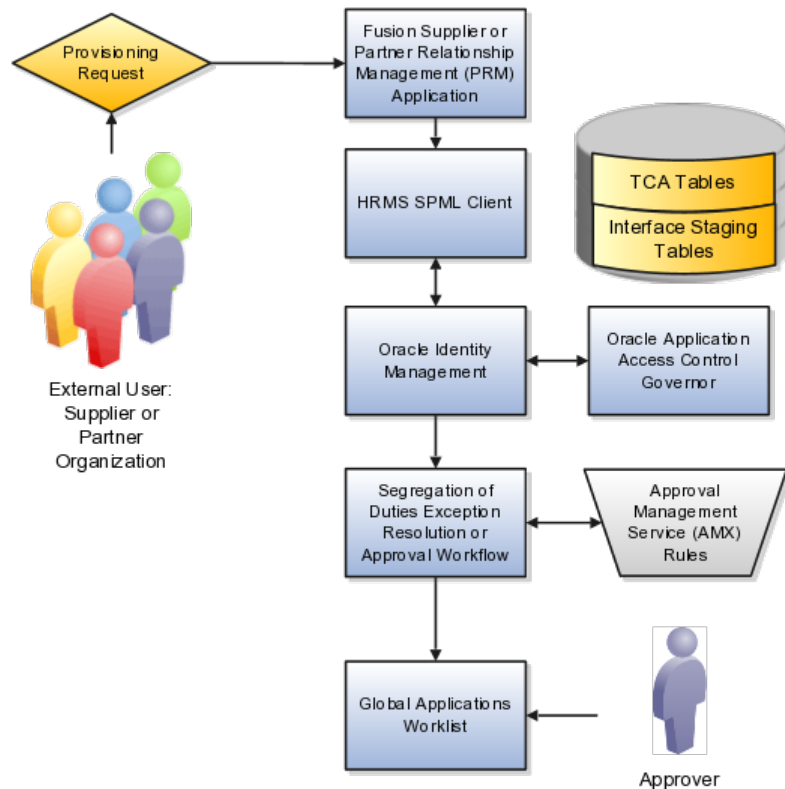
Segregation of duties (SOD) checks occur when roles are assigned to users. The checks are based on Oracle Application Access Controls Governor (AACG) policies in Oracle Enterprise Governance, Risk and Compliance (GRC). The Oracle Identity Management (OIM) integration includes predefined routing rules 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.

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

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

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

Note

When a partner user is provisioned, all employees of the partner enterprise are provisioned. SOD checks occur when an external user requests to join a program, because SOD policies operate across Oracle Fusion Applications, not at the individual level. Supplier or partner company user requests are not approved if there is an SOD conflict against the supplier company.

OIM provides AACG with the details of SOD exception approval workflow. AACG audits the outcome for use in future detective controls and audit processes.

Oracle Application Access Controls Governor

AACG may respond with the following.

- Roles may be provisioned to the external user or its employees because no SOD conflict is found
- SOD conflict is found and request is denied because the relevant SOD policy is to be strictly enforced and no exception approval should be allowed
- SOD conflict is found and the exception to the policy is allowed, so the request goes through additional processing, such as an approval process.

Supplier or Partner Relationship Management responds to an SOD exception by updating Trading Community Model tables with the current state. An enterprise may elect to implement a landing pad that offers external users a means of addressing the SOD problem by providing more information or withdrawing the request.

SOD violation checking occurs during role implementation and provisioning, and can be turned on or off if AACG is provisioned and enabled as part of the Oracle Fusion Applications deployment.

Segregation of Duties Exception Resolution or Approval Workflow

Depending upon status, OIM kicks off an auditable SOD exception resolution workflow. Resolution can be conditional based on approval or requirements such as contracts being met.

If one of the paths for exception resolution is to get an approval, then the SOD exception resolution drives the approval using AMX. Standard AMX rules, not business rules, resolve the approval for the SOD exception, including the following.

- Organizational hierarchies
- Multiple mandatory and optional approvers

- Rerouting and approval delegation

The approver resolution uses AMX Rules Designer to access various user attributes and organizational hierarchies managed in Oracle Fusion Applications repositories. This information is typically not available in OIM or the LDAP identity store repository. Enterprises can define additional approval rules using AMX Thin Client.

The SOD Exception Approver gets a notification through supported channels that a new request is awaiting approval. The approver signs in to the global SOA federated worklist application that aggregates all pending worklist items for the user from all Oracle Fusion applications and logical partitions or pillars of applications. The SOD exception approval tasks show up in the same list.

The SOD exception approval task shows the details of the SPML request and SOD Provisioning results in a page rendered by OIM. The approver may take one of the following actions.

- Approve the request as it is
- Reject the request

If the approver approves the request, OIM sends an `SOD_REMEDIATION_APPROVED` status to the SPML client.

If the approver rejects the request, OIM sends an `SOD_REMEDIATION_REJECTED` status to the SPML client. The provisioning request is considered completed with a failure outcome and the external users is notified. Oracle Fusion Applications updates the Trading Community Model tables with the rejected status

Remediation Task Assignments

The SOD remediation tasks are assigned based on the role being requested.

1. If the role requested is Chief Financial Officer, the SOD remediation task is assigned to the IT Security Manager role.
2. If the SOD violation results from a policy where the SOD control tag is the Information Technology Management business process and the control priority is 1, the SOD remediation task is assigned to Application Administrator role.
3. In all other scenarios, the SOD remediation task is assigned to the Controller role.

For more information about configuring audit policies, see the Oracle Fusion Applications Administrator's Guide.

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

Define Help Configuration

Define Help Configuration: Overview

The Define Help Configuration task list contains tasks that let you set up and maintain Oracle Fusion Applications Help for all users. Use the Set Help Options task to determine if certain aspects of Oracle Fusion Applications Help are available to users and to control how aspects of the help site work. Use the Assign Help Text Administration Duty and Manage Help Security Groups tasks to set up customization of help content.

After performing the help configuration tasks, you can review the predefined help and consider whether to add or customize any content. You can also customize help that is embedded in the application, for example hints and help windows, using other tools such as Oracle JDeveloper and Oracle Composer.

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

Set Help Options

Help Feature Choices and Help Options: Points to Consider

Help feature choices on the Configure Offerings page in the Setup and Maintenance work area control the look and behavior of Oracle Fusion Applications Help, and also determine which help options are available. Help options are setup options on the Set Help Options page.

Local Installation of Help

Select the Local Installation of Help feature choice so that the Define Help Configuration task list appears in your implementation project, and you can select two additional features (Access to Internet-Based Help Features and Help Customization) to control the fields available on the Set Help Options page.

Access to Internet-Based Help Features

Select this feature choice to provide users access to features that involve navigation to sites on the Web. If you select this feature choice, then the Web

Sites Available from Help Site section is available on the Set Help Options page. 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 the Oracle Fusion Applications Technology Library. If you select this feature, then these links open the library on the Oracle Technology Network Web site. If you do not select this feature, then your system administrator must download the library from Oracle Technology Network (<http://www.oracle.com/technetwork/documentation/fusion-apps-doc-1508435.html>) and put all the content from within the extracted folder (for example E28271_01) directly into the appmgr/APPLTOP/fusionapps/applications/ahc/afh/reference/TechLib folder.

Help Customization

Select the Help Customization feature choice if you intend to customize predefined help or add your own files to help. For example, you can add internal policies or procedures as help, and Oracle User Productivity Kit content, if any. 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.

Administering Collaboration Features and Announcements in Help: Points to Consider

Announcements and collaboration features (discussions, ratings and comments) allow users to share information regarding help and the subjects that particular help files cover. The collaboration features are also used elsewhere in Oracle Fusion Applications. Discussions may not be available in Oracle Cloud implementations.

Use the Set Help Options page in the Setup and Maintenance work area to enable the announcements and discussions features and to set options about ratings. When administering these features, consider the purpose of each feature and points that are specific to Oracle Fusion Applications Help.

Announcements

Use announcements to broadcast information to all users of your help site. You can provide information about help, for example new custom help that was recently added, or about anything that users should take note of, for example a change in company policy. Announcements can appear on any of the tabs on the home page of Oracle Fusion Applications Help. You can target specific user groups by posting announcements to specific tabs, for example, posting information related to implementation to the Functional Setup tab.

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

Note

Use the full URL, for example <http://www.oracle.com>, when creating links.

Discussions

Users can use discussions to post questions or comments about subjects covered in specific help files. For example, after reading help on expense reports, users might have questions or comments about company policies or processes for expenses. Other users who later access this help file would benefit from the information in the discussion.

You can set a help option to enable discussions. Each help file would contain a **Discuss** link that all users can use to read discussions about that file. They can also start a discussion topic or post to existing topics. These discussions are visible only to users in your enterprise.

Important

Do not enable discussions until servers for discussions are up and running.

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

Ratings and Comments

Users can rate any help file on a five star system and provide feedback about the content. This information is helpful to other users in deciding which help file to open. Help files with a higher average rating are listed first in help windows, and in the help listings you see as you browse using the help navigators.

The scope of ratings and reviews is limited to your enterprise.

FAQs for Set Help Options

When do I link to the Oracle User Productivity Kit library from the help site?

Provide a link to your Oracle User Productivity Kit (UPK) library if you have UPK licensed and custom UPK content to share with your users. You give them access to a library of custom UPK content in addition to any custom UPK demos that you added to the help site itself. UPK demos that you add as custom help are available only in the See It mode, so the library can include the same demo in other modes. If you have UPK versions earlier than 3.6.1, then you cannot add UPK demos as custom help, so the link is the only way for users to access custom UPK content from the help site.

How can I find the URL to the Oracle User Productivity Kit library?

The URL to enter on the Set Help Options page should be the full path from the Web server where you are hosting your Oracle User Productivity Kit (UPK) content to the index.html file that opens the table of contents for the library, for example, `http://<your domain>.com/UPKcontent/PlayerPackage/index.html`. In this example, you or your UPK administrator would publish one UPK player package that contains all the content to be linked to from Oracle Fusion Applications Help, as well as the index.html file, and place the PlayerPackage folder in a manually created folder called UPKcontent on the Web server.

FAQs for Assign Help Text Administration Duty

Who can add and manage custom help?

Users with the 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.

Decisions to Consider	In This Example
What type of users do you need to limit help access to?	Human resources (HR) specialists

Is there a specific time period for which this access is needed?	No, the help files should always be viewed only by the HR specialists
Where do you want this group to appear in the list of values for help security groups?	First

Define a help security group and assign a duty role to the group.

1. From the Setup and Maintenance work area, find the Manage Help Security Groups task and click **Go to Task**.
2. On the Manage Help Security Groups page, add a new row.
3. Complete the fields, as shown in this table. Leave the start and end dates blank.

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

4. Click **Save**.
5. With your new help security group selected, go to the Associated Roles section and add a new row.
6. Select **PER_HUMAN_RESOURCE_SPECIALIST** as the role name.
7. Click **Save and Close**.

You have created a new lookup code for the Help Security Groups lookup type, which is a standard lookup. The lookup code has the name, meaning, and description that you defined for the help security group.

You have also created a data security policy for the help database resource, specifying that the Human Resource Specialist role can view help that is defined with the HR security group. If you go to the Manage Database Resources and Policies page and find the database resource, or object, ATK_KR_TOPICS, then you can see the policy for the Human Resource Specialist role, with the condition that the column name, SECURITY_CODE, is equal to the value HR.

Help File Customization

Help File Customization: Overview

If you have the appropriate job roles, then you can customize the help files in the help site. Use the Manage Custom Help page to maintain both predefined and custom help files. You can create, duplicate, edit, and delete custom files, or set their status to Active or Inactive. For predefined files, you can only duplicate them or set their status. For each help file, predefined or custom, use help

locations to determine where the help file appears in the application and in the help site. You have various options in how you add custom help, for example by uploading a file or specifying a URL.

Note

To make a copy of all custom help for testing, migration, or other purposes, use the export and import feature in the Setup and Maintenance work area.

There are various ways to access help customization.

- Many help files can be accessed from help windows in the application. If you want to customize help in the context of a help window, for example create a custom help file and add a link to it from a specific help window, then start by opening that help window. When you click the **Manage Custom Help** link, you go to the Manage Custom Help page, and the help location fields are automatically populated with values that correspond to the help window. This way you can easily select existing files to add to the same help location, and when you create a new file, the same help location appears by default.
- Open the Manage Custom Help page directly from the home page of Oracle Fusion Applications Help or from search result pages.
- To edit a specific file, you can either find it in the Manage Custom Help page, or open the file itself and click the **Edit** link.
- Likewise, you can find glossary terms in the Manage Custom Help page, or click the **Glossary** link in the global area to open the Glossary tab, search for the term, and click **Edit**.

Note

When you search in the Manage Custom Help page, make sure that the **Custom Help Only** check box is not selected if you are looking for predefined help.

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

Help Types: Explained

Oracle Fusion Applications Help contains various types of help content, including demos, examples, FAQs, glossary terms, help topics, and PDF guides. A business process or product can be supported by some or all of these help types.

Demo

Demos are Oracle User Productivity Kit (UPK) topics that visually demonstrate how to use the application to complete a short task or portion of a task. Demos can also provide an introduction to complex dashboards and work areas.

Example

Examples provide real use cases of features to illustrate how and when to use the feature, or scenarios to illustrate abstract concepts. Worked examples show exactly what you need to do to achieve a specific result, emphasizing decisions that you make and values that you enter.

FAQ

FAQs, or frequently asked questions, provide brief answers to questions that you might have regarding a task or page. For example, they can briefly explain what a term means, why something happened, how you can perform an action, or what happens if you perform the action.

Glossary

Glossary terms provide definitions for words or phrases used in help. You can search or browse glossary terms in the Glossary tab of Oracle Fusion Applications Help. Where the links are available, you can also see the definition when you hover over the term in help content for other help types.

Help Topic

Help topics explain key concepts, illustrate how application components work together, or assist in decision-making by explaining points to consider or the options you have. Help topics can also provide reference, overview, and other information.

PDF Guide

PDF guides present a collection of help content from the other help types, except demos, in an organized and logical format. For example, there are guides addressing specific business processes and setup offerings. You can see lists of all guides from the **Guides** menu in Oracle Fusion Applications Help.

Help Locations: Explained

Help locations determine where users can find help files, custom or not, from either the application or the help site.

Help locations include:

- Page or section values
- Help hierarchies
- Primary locations

Page or Section Values

The value in the **Page or Section** field on the help customization pages represents where users can click a help icon to open a help window that contains

a link to the help file. In most cases, this value represents a page or region header in the application. Help windows are also available on specific tabs or windows, and in the Setup and Maintenance work area for specific task lists or tasks. You can associate a help file with multiple page or section values, or with none at all.

The page or section value reflects the logical navigation to the help window. For example, **Edit Opportunity page, Revenue tab, Recommendations window** does not mean that the help file is available in three different places. The help icon is in the Recommendations window, which is accessed from the Revenue tab on the Edit Opportunity page.

If the value suggests multiple locations, for example **Create and Edit Opportunity pages**, then the help file is available from the page header of both the Create Opportunity and Edit Opportunity pages. If the page or section value is, for example, a dashboard region that appears in multiple dashboards, then the value does not specify the page name but just the region. The help file is available from that region in multiple dashboards.

Help Hierarchies

Help files are associated with help hierarchies, which are used to categorize help files and aid users in finding help. Each help file can have multiple hierarchies, with at least one of type Business Processes. The business process hierarchy is based on the Business Process Management model. Every page or section value is predefined with a specific business process hierarchy. If you select a page or section without entering a business process hierarchy, the predefined hierarchy appears by default.

The Search by Business Process navigator in the help site is based on the business process hierarchy. For example, if you assign two business process hierarchies to a help file, users can find the file in both locations in the navigator. When the user clicks **More Help** from a help window, all help files assigned to the same business process hierarchy as the page or section value are returned as search results.

Similarly, the Search by Product navigator is based on the Product hierarchy type, in which level 1 is the product family, level 2 is the product, and level 3 is the business activity owned by that product.

The Search by Functional Setup navigator is based on the Functional Setup hierarchy type. The level 1 nodes for this hierarchy are:

- Functional Setup Manager, which includes help about using the Setup and Maintenance work area.
- Guides, which contains level 2 nodes that correspond to business areas and setup offerings. All the user reference and functional setup PDF guides are included.
- Offerings, which contains level 2 nodes for each setup offering, and lower levels for the main task lists in the offerings. Help for the task lists and tasks are included.

The Search by Common Tasks navigator is based on the Welcome hierarchy type. The level 1 nodes represent categories of functional areas common to all users.

Primary Locations

The primary location of a help file designates the hierarchy that is displayed for the help file in search results and within the help content as breadcrumbs. You cannot change the primary location of a help file that came with your help installation. Primary locations of predefined help are based on the business process hierarchy, while custom help files can have primary locations based on hierarchies of any type.

Editing Predefined Help and Glossary Terms: Points to Consider

When you open any predefined help file, including glossary terms, that came with Oracle Fusion Applications Help, you can see an edit option if you have roles allowing edit access. When you edit predefined help, keep in mind:

- What happens to the original help file
- Where predefined help appears
- Considerations specific to glossary terms

What Happens to the Original Files

When you edit predefined help, you are actually creating a new custom help file based on the original file, with the same help locations. The customized version replaces the original, which becomes inactive and hidden from users. You can display both versions by reactivating the original in the Manage Custom Help page.

Note

In the Search Results: Existing Help region on the Manage Custom Help page, there is no option to edit predefined help. You can duplicate a predefined help file, edit the copy, and optionally inactivate the original.

Where Predefined Help Appears

All predefined help comes with preassigned help locations, including at least one based on the hierarchy of type Business Processes. Many also have predefined page or section values that indicate where the help can be accessed from help windows in the application.

To change where predefined help appears, either in the help site navigators or in the application, create a duplicate in the Manage Custom Help page. Change or add help locations to your custom copy, and inactivate the original.

Even though glossary terms do not appear in the help site navigators, you still need to enter at least one help location to categorize the glossary term.

Considerations Specific to Glossary Terms

When you edit a predefined glossary term, the original term becomes inactive. Existing links to the glossary term, from other predefined and custom help files, will automatically point to your custom version. If you later inactivate the custom glossary term, make sure to activate the original term so that the links still work.

Links in Custom Help: Points to Consider

When you create or edit custom help, follow best practices when you include links to help files or other content. If you are working on custom help created by duplicating a predefined help file, then you may see existing links from the original file in the Help Content section. The types of links that you can work with include:

- Related help links
- Standard hypertext links
- Links to documentation library content
- Glossary term links

For all link types, except the standard hypertext links, you must create or edit custom help with a Text or Desktop source type. In other words, you must type the help content directly in the application or use an HTML file that you upload to help. For standard hypertext links, the source type can also be URL.

Related Help Links

Related help is the section at the end of help files that contains links to other help files. The syntax for related help contains a comma-separated list of title IDs that represent help files.

This figure provides an example of related links code.

```
OfaRelatedTopics(CREATE_AUTOMATIC_POSTING_CRITERIA_S_0000,  
JOURNAL_ENTRIES_HOW_THEY_RE_RECORDE_0000)
```

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

Standard Hypertext Links

You can create standard hypertext links to any file or Web site as long as you ensure the stability and validity of the links, including links to other help files,

custom or not. These links can appear anywhere in the body of your help file as long as they come before any related help links.

In the Help Content section, highlight the text that you want to use as link text and click the **Add Link** icon button.

For links to other help files, open the file to which you want to link, and click the **E-Mail** link. Use the URL in the autogenerated e-mail text as the link to the file.

Note

Use the full URL, for example `http://www.oracle.com`, when creating links.

Links to Documentation Library Content

The syntax for links to HTML files in documentation libraries, for example the Oracle Fusion Applications Technology Library, is:

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

WCSUG4636 is the anchor ID and Understanding Tags is the link text. You can:

- Modify the link by replacing the existing anchor ID or editing the link text, or both.
- Remove the link by deleting all the code for it.
- Create links to documentation library content by following the same syntax. These links can appear anywhere in the body of your help file as long as they come before any related help links.

Important

To ensure that you are linking to a supported documentation library, enter anchor IDs only from documentation libraries that are linked from predefined help topics.

Glossary Term Links

Glossary term links provide definitions in a note box when users hover over the term in help files.

This figure shows an example of code for a glossary term link.

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

In this example, `accounting period` is the link text, or glossary term, and `ACCOUNTING_PERIOD_001` is the identifier, or title ID.

- To remove the link but retain the text, delete all the code except the term itself.

- To add glossary term links, you must follow the link syntax and use the correct title ID for the glossary term. You can find title IDs in the search results of the Manage Custom Help page.

Note

If your custom help has glossary terms and the source type is Desktop File, then make sure before uploading that the quotes around the glossary term are actual quotation marks in raw HTML, not ". Otherwise, quotation marks will appear when users view the help file.

Customizing PDF Guides: Worked Example

This example demonstrates how to customize a PDF guide that came with Oracle Fusion Applications Help. This guide is currently not available from any help window in the application.

The following table summarizes key decisions for this scenario.

Decisions to Consider	In This Example
What changes do you need to make to the guide?	Change the title of a chapter and remove a section in that chapter, to hide content about a particular subject
Which help window should the customized guide appear in?	The help window for the entire Welcome dashboard of Oracle Fusion Applications
Which help navigators should the customized guide appear in, and on which node?	Same as the original guide, plus the path associated with the help window
Do you want to limit access to the customized guide?	No, same as the original guide

Edit a copy of the original PDF guide, and use the Manage Custom Help page to replace the original PDF guide with your new file.

Copying and Editing the PDF Guide

1. Open the original PDF guide from the help site and save a copy to your desktop. Leave open the help file for the guide.
2. Using a PDF editor application, change the title of the chapter wherever the chapter title appears. Delete the content you want to hide from users.
3. Make sure that your new PDF guide is less than 6 MB.

Replacing the Original PDF Guide

1. In the help file that you still have open for the original PDF guide, click the **Edit** link.
2. On the Create Help page, use the default values except where indicated.
3. Update the title to the name that you want to display to users.
4. In the **File Name** field, browse for and select your customized guide.

5. Delete any keywords or parts of the description relevant to the content you removed from the PDF guide.
6. Add a help location with the Business Processes hierarchy type and select **Information Technology Management** as the level 1 node, **Manage Enterprise Application Software** as the level 2 node, and **Use Applications** as the level 3 node.
7. Select **Welcome page** in the **Page or Section** column.
8. Click **Save and Close**. The help file for the original PDF guide is automatically set to inactive.

Adding Custom UPK Content to Help: Worked Example

This example demonstrates how to add custom Oracle User Productivity Kit (UPK) topics as demo help files. These help files function like any predefined help file for demos. You can search and include these files in help windows and navigators as you would other help.

In this scenario, you are adding two demos about social networking, to appear in help windows on the Welcome dashboard.

Note

Your demo must be made with UPK 3.6.1 or later to be added as help.

The following table summarizes key decisions for this scenario.

Decisions to Consider	In This Example
What UPK content do you want to add to help?	From a UPK module containing five topics, add two as custom demos on the help site
Which help navigators should each demo appear in, and on which node?	Because the two demos are about social networking: <ul style="list-style-type: none"> • Search by Common Tasks navigator, under the Collaboration node • Search by Business Process navigator, under Information Technology Management - Manage Networking and Communications - Manage Social Networking Capabilities
Which help window should each demo appear in?	On the Welcome dashboard of Oracle Fusion Applications, one demo goes in the help window in the Activity Stream region, and the other in the People Connection region
Do you want to limit access to the help files for the demos?	No
Do you want the help files to appear in the New and Updated pane?	Yes

Generate a report of UPK document IDs, which you will use when creating custom help, to identify the UPK topics that you want to add. Publish the UPK module as a player package, then create custom help for the UPK topics that you want to use as help demos.

Generating a UPK Document ID Report

1. In the UPK Developer, select **Details View**.
2. Right-click any column header, for example Name, and select **Column Chooser**.
3. In the Column Chooser dialog box, click and drag the Document ID column header and drop it after the Name column. Close the Column Chooser dialog box.
4. From the File menu, select to print, and save the output as a Microsoft Excel file to your desktop.

Creating the Player Package

1. From the UPK Developer, make sure that the topics that you want to add as demos have the See It play mode. The topics can also have other modes, but only the See It mode is included in the custom help file.
2. Publish the module, specifying any location for the output and selecting to publish the selection only.
3. In the Formats section of the Publish Content window, select the **Player** check box under the **Deployment** check box group.
4. In the Player section, select the **Include HTML Web Site** check box, to ensure that the custom help file includes a text-only version of the UPK topic.
5. Finish the publishing process, after optionally setting other options.
6. Navigate to the location where you specified the output to be generated.
7. In the Publishing Content folder, copy the PlayerPackage folder and add it to the web server where you store UPK content.

Creating Custom Help for Demos

1. Open the help window in the Activity Stream region on the Welcome dashboard of Oracle Fusion Applications, and click **Manage Custom Help**.
2. On the Manage Custom Help page, the page or section and hierarchy values are populated with the values for the Activity Stream region.
3. Click **Create**.
4. On the Create Help page, complete the fields in the General Information section, as shown in this table. Use the default values except where indicated.

Field	Value
Title	The name of the UPK topic.
Source Type	Oracle User Productivity Kit
File Location	The full URL of the player package folder on the Web server, for example, <code>http://<your domain>.com/UPKcontent/PlayerPackage.</code>

Document ID	The document ID of the UPK topic to add to the help window in the Activity Stream region. You can copy and paste this ID from the Microsoft Excel file that you generated earlier.
Help Type	Demo
Help Security Group	Unsecured
Keywords	Terms relevant to the demo.
Description	Summary of the demo.
Include in New and Updated pane	Selected

The Help Location section contains values for the help window in the Activity Stream region. This help file will also appear in the Search by Business Process navigator under this predefined hierarchy.

5. Click **Save and Close**.
6. On the Manage Custom Help page, open the help locations for the help file that you just created.
7. Add a help location with the Welcome hierarchy type and select **Collaboration Features** as the level 1 node.
8. Add another help location with the Business Processes hierarchy type and select **Information Technology Management** as the level 1 node, **Manage Networking and Communications** as the level 2 node, and **Manage Social Networking Capabilities** as the level 3 node.
9. Click **Save and Close**.
10. Starting at the Connections region, repeat steps 1 to 9 for the other UPK topic that you want to add.

FAQs for Help File Customization

How can I restrict help content to specific user roles?

When you create or edit help, select a help security group that represents the set of roles that you want to have access to the help. If you do not see the Security Group field, then your administrator has not selected the Custom Help Security feature choice. The Unsecured group has no associated roles, so anyone can view the help. The predefined Secured group includes all internal employees and contingent workers, unless this group has been edited. You can create security groups and associate roles using the Manage Help Security Groups page, which you can access by starting in the Setup and Maintenance Overview page and searching for the Manage Help Security Groups task. Your new security groups are immediately available for use to secure new or edited help files.

Why can't I select and add help to a location?

You must specify a page or section to add the existing help to. To ensure that help is added to the correct help window, go to the page or section in the

application, click the **Help** icon, and click the **Manage Custom Help** link in the help window. Alternatively, in the Manage Custom Help page, search for at least a page or section and a level 1 value for the Business Processes hierarchy type before selecting the **Select and Add** option.

You cannot select and add help to a particular hierarchy, on the Manage Custom Help page, without a page or section. To add just a hierarchy, search for the help file, add a new help location, and specify only the hierarchy information.

What happens to custom help when a help patch is applied?

Oracle Fusion Applications Help patches update all help files, both active and inactive, except custom help. Custom help files are not affected by patches. Consider reviewing inactive files to see if you want to activate the updated version, or to make similar edits to the custom versions of those files, if any.

Embedded Help Customization

Customizing Embedded Help: Highlights

You can customize help that is embedded in the application, for example hints and help windows, for all users of Oracle Fusion Applications.

Embedded help customization is fully described in the Oracle Fusion Applications Extensibility Guide.

- Edit, create, or delete hint text that appears on hover over buttons, links, icons, or tab titles.

See: Customizing or Adding Bubble Embedded Help

- Edit, create, or delete other types of embedded help. Refer to the Customizing or Adding Static Instructions, In-Field Notes, and Terminology Definitions section.

See: Oracle Fusion Applications Extensibility Guide

Maintain Common Reference Objects

Maintain Common Reference Objects: Overview

The Maintain Common Reference Objects task list contains Oracle Fusion Middleware Extensions for Applications (Applications Core) tasks that support implementation of common behaviors, such as data security or reference data sets.

Use this task list to manage common reference objects that are defined centrally and shared across applications, in addition to those that are specific to Applications Core functionality. You can access this task list by starting in the Setup and Maintenance Overview page and searching for common reference object task lists.

Note

Offerings also include application-specific tasks for managing Applications Core objects. For example, the Financials offering includes tasks such as Manage Receivables Descriptive Flexfields, and Manage Receivables Lookups.

For more information on configuring custom objects, see the Oracle Fusion Applications Extensibility Guide.

To make the Maintain Common Reference Objects task list available in your implementation project, go to **Setup and Maintenance Overview - Configure Offerings**, and for a specific offering, select the Maintain Common Reference Objects feature choice.

Define Application Taxonomy

Application Taxonomy: Highlights

Application taxonomy is the organization of Oracle application components and functions in a hierarchical structure, from product lines to logical business areas. This hierarchy represents a breakdown of products into units based on how applications are installed and supported. Maintain this hierarchy on the Manage Taxonomy Hierarchy page, which you can access by starting in the Setup and Maintenance Overview page and searching for the Manage Taxonomy Hierarchy task.

A detailed introduction to application taxonomy is provided in the Oracle Fusion Applications Developer's Guide.

Hierarchy

- The application taxonomy hierarchy contains various levels and types of nodes, or modules.

See: Characteristics of the Level Categories

See: Benefits of a Logical Hierarchy

Usage

- Use application taxonomy to understand relationships among applications and between an application and its files. This information is helpful in managing various phases of the product lifecycle.

See: How to Manage the Lifecycle

Modules in Application Taxonomy: Explained

A module is any node in the application taxonomy hierarchy. The top level of the hierarchy is product line, followed by product family, application, and logical business area. There can be multiple levels of logical business areas, with one or more nested within a parent logical business area.

Product Line

A product line is a collection of products under a single brand name, for example, Oracle Fusion.

Product Family

A product family is a collection of products associated with a functional area that may or may not be licensed together as a single unit, for example Financials.

Application

An application is a single product within a product family, containing closely related features for a specific business solution, for example General Ledger.

Logical Business Area

A logical business area is a collection of business object definitions organized into a logical grouping. It contains the model objects, services, and UI components for those business objects. Logical business areas have their own hierarchy levels and in some cases can be two or three levels deep. Each leaf node has at least one business object and service, up to a maximum of four business objects and associated services. A logical business area with more than four business objects are further refined with child logical business area levels. Each of these parent-child levels is represented by a directory in the physical package hierarchy.

Managing Modules in Application Taxonomy: Points to Consider

Manage modules on the Create Child Module or Edit Module page, which you can access by starting in the Setup and Maintenance Overview page and searching for the Manage Taxonomy Hierarchy task. When you create a module, it is a child of the currently selected node in the application taxonomy hierarchy. This determines which values are available, for example for module type. Once created, you cannot delete the module or move it elsewhere in the hierarchy. As you create or edit modules, consider the following points regarding specific fields.

Identifiers

Module ID is the unique primary key for nodes in the taxonomy table. When you create a module, an ID is automatically generated. Once the module is created, you cannot update the ID.

Module key and alternative ID are additional identifiers of the module, presented in a way that is easier to read than the module ID. The module key is a string identifier, for example AP for the Oracle Fusion Payables application. The alternative ID is a numeric identifier, for example 1 for the Oracle Fusion product line. These IDs are provided for the product line, product family, and application modules, but you can optionally add them for logical business areas and new custom modules.

Note

Do not change the module key or alternative ID for predefined modules.

The product code is relevant only to application and logical business area modules. You can leave the field blank for other module types. The product code for applications is the short name that can be displayed in lists of application values, for example FND for Oracle Fusion Middleware Extensions for Applications.

Names

Module name is the logical name for the module and is always available. The name must be unique among nodes in the same hierarchy level with the same parent, but try to make it as unique in the whole hierarchy as possible.

The user name and description can appear to users in other parts of Oracle Fusion Applications, so make sure that the values are something that users know to represent the module.

Usage Types

Though you can update the usage type to reflect the current state of the module, just doing so does not affect the actual state. For example, setting a module as installed does not mean it is actually installed if the installation itself has not taken place. Installation refers to operations related to laying down all the components needed to create an Oracle Fusion Applications environment, while

deployment is the process that starts the managed servers and clusters and facilitates the actual use of product offerings. A licensed module is available for installation and deployment, and a deployed module is considered actively used when actually used by users.

Seed Data

If seed data is allowed, then seed data such as flexfields and lookups can be extracted for the module using seed data loaders. By default, extract is allowed for all predefined modules of type application and logical business area.

Associations

You can associate a logical domain to modules of type product family, as well as one or more enterprise applications to modules of type application. This association represents the relationship between the taxonomy modules and the corresponding domain and enterprise applications stored in the Oracle Fusion Applications Functional Core (ASK) tables.

Define Reference Data Sharing

Reference Data Sharing: Explained

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

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

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

Note

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

Reference Data Sets: Explained

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

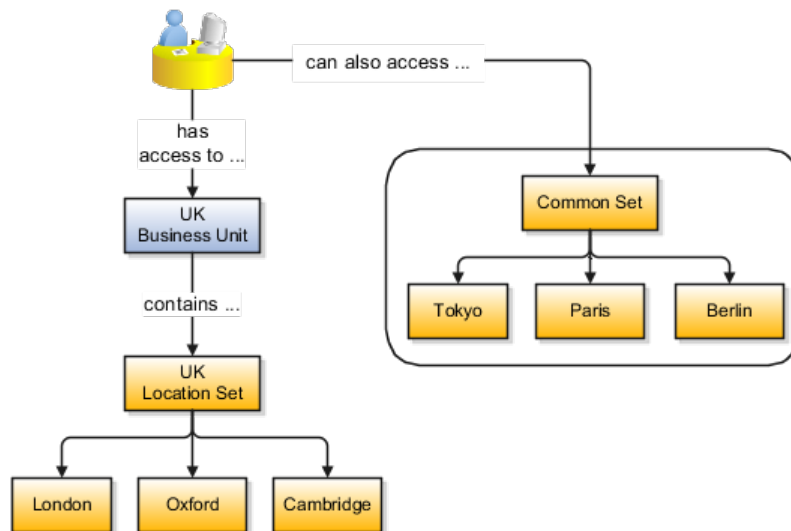
Consider the following scenario.

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

Partitioning

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

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



Assigning Reference Data Sets to Reference Objects: Points to Consider

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

Determinant Types

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

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

Determinant

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

Reference Groups

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

Note

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

Define Lookups

Lookups: Explained

Lookups are lists of values in applications. You define a list of values as a lookup type consisting of a set of lookup codes, each code's translated meaning, and optionally a tag. End users see the list of translated meanings as the available values for an object.

Lookups provide a means of validation and lists of values where valid values appear on a list with no duplicate values. For example, an application might store the values Y and N in a column in a table, but when displaying those values in the user interface, Yes or No (or their translated equivalents) should be available for end users to select. For example, the two lookup codes Y and N are defined in the REQUIRED_INDICATOR lookup type.

In another example, a lookup type for marital status has lookup codes for users to specify married, single, or available legal partnerships.

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

In this case, tags are used for localizing the codes. All legislations list Married and Single. Only the Dutch legislation lists Registered Partner. And all legislations except France and Australia also list Domestic Partner.

When managing lookups, you need to understand the following.

- Using lookups in applications
- Customization levels
- Accessing lookups
- Enabling lookups
- The three kinds of lookups: standard, common, and set enabled

Using Lookups in Applications

Use lookups to provide validation or a list of values for a user input field in a user interface.

An example of a lookup used for validation is a flexfield segment using a table-validated value set with values from a lookup type. An example of a lookup in a list of values is a profile option's available values from which users select one to set the profile option. Invoice Approval Status gives the option of including payables invoices of different approval statuses in a report. The lookup code values include All so that users can report by all statuses: Approved, Resubmitted for approval, Pending or rejected, and Rejected.

Customization Level

The customization level of a lookup type determines whether the lookups in that lookup type can be edited. This applies data security to lookups.

Some lookup types are locked so no new codes and other changes can be added during implementation or later, as needed. Depending on the customization level of a lookup type, you may be able to change the codes or their meanings. Some lookups are designated as extensible, so new lookup codes can be created during implementation, but the meanings of predefined lookup codes cannot be modified. Some predefined lookup codes can be changed during implementation or later, as needed.

The customization levels are user, extensible, and system. The following table shows which lookup management tasks are allowed at each customization level.

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

Predefined data means LAST_UPDATED_BY = SEED_DATA_FROM_APPLICATION.

If a product depends on a lookup, the customization level should be system or extensible to prevent deletion.

Once the customization level is set for a lookup type, it cannot be modified. The customization level for lookup types created using the Define Lookups page is by default set at the User level.

Standard, Common, and Set-Enabled Lookups

The available kinds of lookups are as follows.

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

Common	Legacy lookups
--------	----------------

Standard lookups are the simplest form of lookup types consisting only of codes and their translated meaning. They differ from common lookups only in being defined in the standard lookup view.

Common lookups exist for reasons of backward compatibility and differ from standard lookups only in being defined in the common lookup view.

Set enabled lookup types store lookup codes that are enabled for reference data sharing. At runtime, a set-enabled lookup code is visible because the value of the determinant identifies a reference data set in which the lookup code is present.

Accessing Lookups

Standard, set-enabled, and common lookups are defined in the Standard, Set-enabled, and Common views, respectively. Applications development may define lookups in an application view to restrict the UI pages where they may appear.

In lookups management tasks, lookups may be associated with a module in the application taxonomy to provide a criteria for narrowing a search or limiting the number of lookups accessed by a product specific task such as Manage Purchasing Lookups.

Enabling Lookups

A lookup type is reusable for attributes stored in multiple tables.

Enable lookups based on the following.

- Selecting an **Enabled** check box
- Specifying an enabled start date, end date, or both
- Specifying a reference data set determinant

If you make changes to a lookup, users must sign out and back in before the changes take effect. When defining a list of values for display rather than validation, limit the number of enabled lookup codes to a usable length.

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.

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

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

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

Understanding the Resulting Data Entry List of Values

Users need to respond to a process question by indicating whether to stop it, use caution, go ahead, or complete it urgently.

The list of values for the COLORS lookup type includes the meanings for the enabled codes.

Displayed Value	Hidden ID
Stop	RED
Caution	YELLOW
Go	GREEN

Analysis

The BLUE lookup code was not enabled and does not appear in the list of values. The display sequence of values in the list of values is alphabetical unless you enter a number manually to determine the order of appearance. Number 1 indicates the value listed first in the list of values.

Note

Only lookups that are enabled and active, meaning between start and end dates, are visible.

When users enter one of the values from the list of values for the lookup type COLORS, the transaction table records the lookup code. In this example, the code is stored in the Status column

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

The status for one user is BLUE because at the time they entered a value, BLUE was enabled. Disabling a lookup code does not affect transaction records in which that code is stored. Data querying and reporting have access to disabled lookup codes in transaction tables.

Managing Set-Enabled Lookups: Examples

Creating a new set-enabled lookup is similar to creating a standard lookup with the addition of specifying a reference data set determinant for the lookup codes.

Note

You can only create or edit the lookup codes for a particular lookup type if its customization level supports it.

The reference data set for a set-enabled lookup code is part of its foreign key. This is unlike other set-enabled entities.

Selecting a Reference Group for a Set-Enabled Lookup Type

By specifying a reference group for a set-enabled lookup type you indicate which reference data set assignments are available for its lookup codes. For example a COLORS lookup type might be set enabled for a Countries reference group that includes the US and EU reference data set assignments.

Selecting a Reference Data Set for a Set-Enabled Lookup

The reference data set determines which lookup code is included in the list of values. If a COLORS lookup type contains a RED, YELLOW, ORANGE, and GREEN lookup code, you can enable one RED lookup as coming from the US reference data set and another RED lookup as coming from the EU reference data set with divergent meanings.

Reference Data Set	Lookup Code	Lookup Meaning
US	RED	Red
US	YELLOW	Yellow

US	GREEN	Green
EU	RED	Rouge
EU	ORANGE	Orange

In addition to divergent meanings for lookup codes based on associated reference data set, some lookup codes may be unique to one or another reference data set as the ORANGE lookup is to the EU reference data set in this example.

In another example, a lookup type called HOLD_REASON provides a list of reasons for applying a hold to a contract renewal. Reference data sets determine which codes are included in the hold reason list of values.

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

Using the Manage Set Assignments task, you have defined assignments that designate the China business unit to refer to the CHINA and the US business unit to refer to the US and all business units to refer to the COMMON set. When end users place a contract hold in the US business unit, only the three reason codes in US_SET are available. When placing a contract hold in the China business, only the two codes in China_SET are available.

FAQs for Define Lookups

How can I edit lookups?

You can edit the existing lookup codes of a lookup type or add new lookup codes on the Define Lookups pages, which you can access by starting in the Setup and Maintenance work area and searching for lookup tasks. You can edit the existing lookup codes of a lookup type, or add new lookup codes to a lookup type, if the customization level for the lookup type supports editing.

Why can't I see my lookup types?

Lookups are listed by lookup type. Typically lookup types are managed using tasks that handle a group of related lookups, such as Manage Geography Lookups. Each task gives you access only to certain lookup types. The generic tasks provide access to all lookup types of a kind, such as all common lookups using the Manage Common Lookups task.

If existing lookups are not available to the tasks of the Define Lookups activity, they may be validated for use in a lookup view that is not central to all

applications or whose owning application has not been specified in a lookup view.

Lookups can only be managed in the Define Lookups tasks if the lookup's view application is the standard lookups view, common lookups view, or set-enabled lookups view. Lookups defined in an application view can only be managed by following instructions provided by the owning application.

Note

A lookup type and its codes can only be defined in one lookup view.

What's the difference between a lookup type and a value set?

A lookup type consists of lookup codes that are the values in a static list of values. Lookup code validation is a one to one match.

A table-validated value set can consist of values that are validated through a SQL statement, which allows the list of values to be dynamic.

Tip

A table validated value set can be defined based on any table, including the lookups table. This allows a lookup type to be made into a table-validated value set that can be used in flexfields.

Area of Difference	Lookup Type	Value Set
List of values	Static	Dynamic if Table validation type
Validation of values	One to one match of meaning to code included in a lookup view, or through the determinant of a reference data set	By format or inclusion in a table
Format type of values	char	varchar2, number, and so on
Length of value	Text string up to 30 characters	Any type of variable length from 1 to 4000
Duplication of values	Never. Values are unique.	Duplicate values allowed
Management	Managed by both administrators and end-users, except system lookups or predefined lookups at the system customization level, which cannot be modified.	Maintained by administrators, except some product flexfield codes, such as GL for Oracle Fusion General Ledger, which are maintained by end users

A lookup type cannot make use of a value from a value set.

Value sets can make use of standard, common, or set-enabled lookups.

Both lookup types and value sets are used to create lists of values from which users select values.

What's a lookup tag used for?

Tags on lookup codes allow you to add a label to your lookup codes.

Lookup tags are unvalidated and uninterpreted by lookups. A tag can be used to categorize lookups based on facilitating searches or guiding how a lookup should be used.

Document what the tag on a lookup represents and how to use it.

Define ISO Reference Data

Defining Currencies: Points to Consider

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

Currency Codes

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

Date Ranges

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

Symbols

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

Euro Currency Derivation: Explained

Use the Derivation Type, Derivation Factor, and Derivation Effective Date fields to define the relationship between the official currency (Euro) of the European

Monetary Union (EMU) and the national currencies of EMU member states. For each EMU currency, you define its Euro-to-EMU fixed conversion rate and the effective starting date.

Note

If you need to use a different currency code for Euro, you can disable the predefined Euro currency and create a new one.

Derivation Type

The **Euro currency** derivation type is used only for the Euro, and the **Euro derived** derivation type identifies national currencies of EMU member states. All other currencies do not have derivation types.

Derivation Factor

The derivation factor is the fixed conversion rate by which you multiply one Euro to derive the equivalent EMU currency amount. The Euro currency itself should not have a derivation factor.

Derivation Effective Date

The derivation effective date is the date on which the relationship between the EMU currency and the Euro begins.

Natural Languages: Points to Consider

Natural languages are all the languages that humans use, written and spoken. If a language is enabled, then users can associate it with entities, for example as languages spoken by sales representatives. When managing natural languages, consider tasks to perform and best practices for entering particular values.

Tasks

Once you add a language, it cannot be deleted, just disabled. You can optionally associate natural languages with International Organization for Standardization (ISO) languages and territories, just for reference.

Values

When you create a natural language, use the alpha-2 ISO code as the language code, or, if not available, then alpha-3. If the language is not an ISO language, then use **x-** as a prefix for the code, for example **x-ja** for a Japanese dialect. Use the **sgn** code of ISO-639-2 for sign languages, followed by territory code, for example **sgn-US** for American Sign Language. You can also use Internet Assigned Numbers Authority (IANA) language tags.

The natural language description should be the language name with territory name in parenthesis where needed, for example **English (Australia)** and **English (Canada)**.

FAQs for Define ISO Reference Data

When do I create or edit territories?

Edit territory descriptions to determine how they are displayed in lists of country values throughout Oracle Fusion Applications. The predefined territories are all countries from the International Organization for Standardization (ISO) 3166 standard. You usually would not edit territory names or codes.

Do not edit National Language Support (NLS) territory codes, which are identifiers used in the system, unless you need to change the association between ISO and system territory. You usually would not edit the default currency, which is the value that defaults in the **Currency** field in Oracle Fusion Applications user preferences after the user first selects a territory.

Create territories if new countries emerge and the system has not yet been patched with the latest ISO country values.

When do I create or edit industries?

Edit industry descriptions to determine how they are displayed in Oracle Fusion Applications. You usually would not edit industry names, which are from the North American Industry Classification System (NAICS). Enabled industries are mainly used in the context of customization, though these values can also appear in any application.

Create industries if you have particular ones you need, for example for customization, that are not included in the NAICS standard.

When do I associate industries with territories?

Optionally associate industries with territories to provide an industry in territory value, used for customization. For example, administrators can customize a page in one way for users within an industry in one country, and another way for users within the same industry in another country. The administrator would select the appropriate industry in territory value to set the customization context.

When do I create or enable currencies?

Create currencies to use, for example for reporting purposes, if they are not already provided. All currencies from the International Organization for Standardization (ISO) 4217 standard are provided.

Enable any currency other than USD for use in Oracle Fusion Applications, for example for displaying monetary amounts, assigning to sets of books, entering transactions, and recording balances. Only USD is enabled by default.

What's the difference between precision, extended precision, and minimum accountable unit for a currency?

Precision is the number of digits to the right of the decimal point used in regular currency transactions. Extended precision is the number of digits to the right of the decimal point used in calculations for this currency, and it must be greater than or equal to the standard precision. For example, USD would have 2 for precision because amounts are transacted as such, for example \$1.00. For calculations, for example adding USD amounts, you might want the application to be more precise than two decimal digits, and would enter an extended precision accordingly.

Note

Some applications use extended precision. Others, such as Oracle Fusion General Ledger, do not.

Minimum accountable unit is the smallest denomination for the currency. For example, for USD that would be .01 for the cent. This unit does not necessarily correspond to the precision for all currencies.

What's a statistical unit currency type?

The statistical unit currency type is used only for the Statistical (STAT) currency. The Statistical currency is used to record statistics such as the number of items bought and sold. Statistical balances can be used directly in financial reports, allocation formulas, and other calculations.

When do I create or edit ISO languages?

You can edit the names and descriptions of International Organization for Standardization (ISO) languages to determine how they are displayed in lists of ISO language values in Oracle Fusion Applications. The ISO languages are from the ISO 639 standard. If there were changes to the ISO standard and the system has not yet been patched with the latest ISO values, you can update the ISO alpha-2 code or add languages as needed.

When do I edit languages?

Installed languages automatically appear on the Manage Languages page, so you do not manually enter newly installed languages. This page contains all languages available for installation and translation in Oracle Fusion Applications. Each dialect is treated as a separate language. The language codes and names are values used by the system.

You generally would not edit any of the detailed fields unless you really need to and know what they are.

When do I create or edit time zones?

Though all standard time zones are provided, optionally enable only a subset for use in lists of time zone values in Oracle Fusion Applications. You can add time

zones if new zones became standard and the system has not yet been patched with the latest values.

Manage Data Security Policies

Data Security in the Security Reference Implementation: Explained

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

The security implementation of an enterprise is likely a subset of the reference implementation, with the enterprise specifics of duty roles, data security policies, and HCM security profiles provided by the enterprise.

The business objects registered as secure in the reference implementation are database tables and views.

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

Data Security Policies in the Security Reference Implementation

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

Tip

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

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

Warning

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

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

may have a different risk tolerance than assumed by the security reference implementation.

HCM Security Profiles in the Security Reference Implementation

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

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

Data Roles

The security reference implementation includes no predefined data roles to ensure a fully secured initial Oracle Fusion Applications environment.

The security reference implementation includes data role templates that you can use to generate a set of data roles with entitlement to perform predefined business functions within data dimensions such as business unit. Oracle Fusion Payables invoicing and expense management are examples of predefined business functions. Accounts Payable Manager - US is a data role you might generate from a predefined data role template for payables invoicing if you set up a business unit called US.

HCM provides a mechanism for generating HCM related data roles.

Data Security: Explained

By default, users are denied access to all data.

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

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

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

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

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

Data security consists of privileges conditionally granted to a role and used to control access to the data. A privilege is a single, real world action on a

single business object. A data security policy is a grant of a set of privileges to a principal on an object or attribute group for a given condition. A grant authorizes a role, the grantee, to actions on a set of database resources. A database resource is an object, object instance, or object instance set. An entitlement is one or more allowable actions applied to a set of database resources.

Data is secured by the following means.

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

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

Data Security Policies

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

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

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

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

- Table or view
- Entitlement (actions expressed by privileges)

- Instance set (data identified by the condition)

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

Note

Some data security policies are not defined as grants but directly in applications code. The security reference manuals for Oracle Fusion Applications offerings differentiate between data security policies that define a grant and data security policies defined in Oracle Fusion applications code.

A business object participating in a data security policy is the database resource of the policy.

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

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

Important

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

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

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

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

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

Data Roles

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

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

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

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

A data role may be granted entitlement over a set people.

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

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

HCM Security Profiles

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

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

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

Masking and Encryption

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

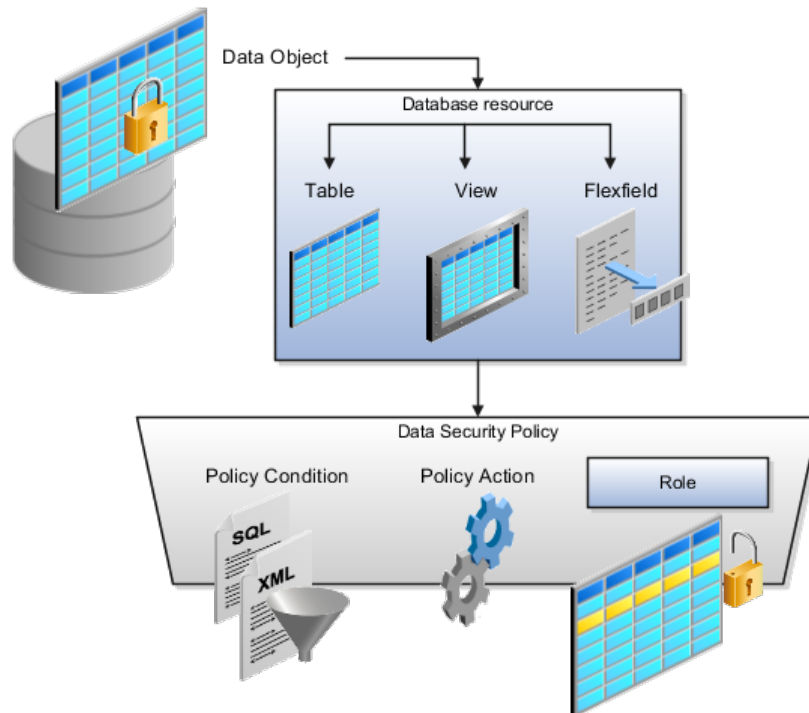
Database Resources and Data Security Policies: How They Work Together

A data security policy applies a condition and allowable actions to a database resource for a role. When that role is provisioned to a user, the user has access

to data defined by the policy. In the case of the predefined security reference implementation, this role is always a duty role. Data roles generated to inherit the job role based on data role templates limit access to database resources in a particular dimension, such as the US business unit.

The database resource defines and instance of a data object. The data object is a table, view, or flexfield.

The following figure shows the database resource definition as the means by which a data security policy secures a data object. The database resource names the data object. The data security policy grants to a role access to that database resource based on the policy's action and condition.



Database Resources

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

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

Data Security Policies

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

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

Note

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

Actions

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

Conditions

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

Tip

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

Securing Data Access: Points to Consider

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

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

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

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

- Specific rows of data (data object) identified by primary key

- Groups of data (instance set) based on a predicate that names a particular parameter
- Data objects or instance sets based on runtime user session variables

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

Grants

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

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

Grants control which data a user can access.

Note

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

A grant logically joins a user or role and an entitlement with a static or parameterized object instance set. For example, `REGION='WEST'` is a static object instance set and `REGION=&GRANT_ALIAS.PARAMETER1` is a parameterized object instance set. In the context of a specific object instance, grants specify the allowable actions on the set of accessible object instances. In the database, grants are stored in `FND_GRANTS` and object instance sets are stored in `FND_OBJECT_INSTANCE_SETS`. Object access can be tested using the privilege check application programming interface (API).

Securing a Business Object

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

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

Note

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

Data Role Templates: Explained

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

Note

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

The following attributes define a data role template.

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

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

Note

Abstract, job, and data roles are enterprise roles in Oracle Fusion Applications. Oracle Fusion Middleware products such as Oracle Identity Manager (OIM) and Authorization Policy Manager (APM) refer to enterprise roles as external roles. Duty roles are implemented as application roles in APM and scoped to individual Oracle Fusion Applications.

The dimension expresses stripes of data, such as territorial or geographic information you use to partition enterprise data. For example, business units are a type of dimension, and the values picked up for that dimension by the data role template as it creates data roles are the business units defined for your enterprise. The data role template constrains the generated data roles with grants of entitlement to access specific data resources with particular actions. The data role provides provisioned users with access to a dimensional subset of the data granted by a data security policy.

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

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

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

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

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

Making Changes To Data Role Templates

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

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

Making Changes to Dimension Values

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

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

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

Define Document Sequences

Document Sequences: Explained

In Oracle Fusion Applications, each business document or business event is uniquely identified by a document sequence number that you assign to it. However, the document sequencing feature must be turned on (enabled) on the business document or event to allow the assignment. For example, if document sequencing is enabled, you can assign a document sequence number to an invoice that gets generated in response to a purchase order. You can use document sequences as a proof to track successfully executed transactions as well as failed transactions. Additionally, a document sequence helps in generating an audit trail, which can be used to identify how a particular transaction passed through various applications.

Document sequencing can be managed automatically, manually, and gaplessly.

Note

Plan your document sequencing carefully before you use the options available in the application to apply sequence numbers. Avoid changes to the options after you saved your work on the Manage Document Sequences and Manage Document Sequence Categories pages.

Automatic Sequencing

Automatic document sequencing assigns a unique number to each document as it is generated, and this unique number is stored in the database. The numbering is sequential by date and time of creation. If you define a sequence to automatically number documents, you can provide an initial value to begin the sequence. In absence of a custom value, the default value 1 is used.

Manual Sequencing

Manual sequencing requires you to assign a unique number to each document before it is generated. In manual sequencing, the numerical ordering and completeness of a transaction is not enforced. Users can skip or omit numbers when entering the sequence value. However, each time that a number is assigned, the application validates its uniqueness.

Gapless Sequencing

Gapless sequencing is similar to automatic sequencing. It automatically generates a unique number for each document, but does that only for successfully generated documents. As a result, the sequence is maintained for all the documents that are generated, and no sequence numbers are lost due to incomplete or failed document generation.

Important

Use this type of sequencing only if necessary because it may affect the performance of the system and slow down transaction processing.

Document Sequence Categories: Explained

A document sequence category is a set of documents that share similar characteristics and that are formed into a logical group. Document sequence categories simplify the task of assigning number sequences to specific documents. Instead of assigning a number to each document, you assign a document sequence to one or more document sequence categories. The document sequence category automatically takes care of numbering the documents.

A document sequence category identifies the database table that stores documents resulting from transactions that your users enter. When you assign a sequence to a category, the sequence numbers the documents that are stored in a particular table. You must create document sequence categories to be able to manage the task of assigning document sequences.

Restriction

Once a document sequence category is created, you cannot change the application, the category code, or the table name. Therefore, carefully consider these details and plan your document sequencing requirement before you begin working with the application.

Once you create a document sequence category, it is available for use under the **Document Sequences: Assignments** section on the Manage Document Sequences page. The **Category** field contains the name of the document sequence category. After you create a document sequence, you can assign it to a document sequence category.

Document Sequences: Points to Consider

Sequencing documents is an important business and legal requirement. Certain aspects of the defining process are permanent and cannot be modified later. Therefore, it is important that you first decide the appropriate document sequence to use for a set of documents. You must also decide beforehand the type of document sequencing, because you are not allowed to switch to other types once a sequence is assigned to a document sequence category. Make a note of the details such as the document sequence and document sequence category so that you can refer to them at a later point in time. Also note if there are any restrictions or configuration prerequisites before you define document sequencing.

Note

Products that implement document sequencing have specifications about its usage. Refer to the corresponding product documentation for specific details and also to determine if there are any restrictions or configuration prerequisites.

Creating and Editing Document Sequences

You can create document sequences that are automatic, manual, or gapless, depending on the business or legal requirement. By default, the current date is considered as the start date. If the end date is left blank, it means that the sequence definition never expires. Among the several options used in creating and editing document sequences, the following options are functionally more important and therefore need to be carefully determined:

- **Determinant Type:** Select to limit the document sequencing activity to certain documents that belong to a specific business entity, such as Ledger, Tax Registration, and so on.
- **Initial Value:** Enter a value for the first document in your sequence. This field applies only to sequences with automatic or gapless numbering types. Sequence numbers should not be greater than eight digits. If you leave this field blank, the first document is automatically assigned a value of 1. Once a document sequence is defined, you cannot change this initial value.

Creating and Editing Document Sequence Categories

Document sequence categories are defined to make it easy to assign document sequence definitions to a group of documents instead of to individual documents. Each document sequence category is mapped to a specific table, where the documents belonging to that category are stored. The table must already be enabled for document sequencing. When specifying the table, you must consider the following points:

- When the sequential numbering feature checks for completeness or generates a report, it locates the category's documents in the table.
- You can select only tables belonging to the application associated with the category.
- Once a category is defined, you cannot change the choice of table.

Assigning Document Sequences

Identify the documents to be numbered before assigning them a document sequence. For each document sequence, there can be only one active assignment to a document sequence category, a method code, and a determinant value (if applicable). As part of the assignment, specify whether the document is created automatically (for example, due to a batch process, or manually through a form). If you do not specify an end date, the assignment continues to remain active throughout the process cycle. If a determinant type was specified for the document sequence, then enter a specific determinant value related to the selected determinant type.

At runtime, when users create documents, the document sequence to be assigned is determined by finding the active assignment that matches the correct combination of category, numbering method, and the date range containing the transaction date.

Auditing Document Sequences

You can audit document sequences, if required, to provide an audit trail of the document sequences used in a specific product. However, before enabling the audit functionality for a document sequence, you must have created an audit table for the specific document sequence, using appropriate details. Enabling the audit functionality is permitted only for newly created document sequences. You cannot audit document sequences that are already in use by a specific product.

For more information about defining a document sequence audit table, see the Oracle Fusion Applications Developer's Guide.

Define Trees

Trees: Overview

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

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

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

Tree Structures

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

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

Tree Versions

A tree is created having only one version. However, users can create more than one tree version depending on the need, and they can make changes to those versions. Depending on varying requirements, users can create one or more tree

versions and publish all of them or some of them by making the versions active at the same time. Similar to any other version control system, versions of trees are maintained to keep track of all the changes that a tree undergoes in its life cycle.

Tree Labels

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

Manage Tree Structures

Tree Structures: Explained

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

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

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

Tree Structure Definition: Points to Consider

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

Tree Node Selection

The **Tree Node** table displays data in nodes that exist in the data hierarchy. You must select the correct and most appropriate tree node table to be able to define the tree structure, based on the tree hierarchy you want to establish. This selection also affects the level of security that is set on a tree node and its child entities.

Tree Sharing Mode

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

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

Creation Mode

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

Customization

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

Multiple Tree Versions

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

Managing Tree Structures: Points to Consider

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

Creating and Editing Tree Structures

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

Note

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

Setting Status

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

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

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

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

Tree Structure Audit Results: Explained

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

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

Running an Audit

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

Validation Details

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

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

<p>Available Label Data Sources</p>	<p>On the Manage Tree Structures: Specify Data Sources page, if a labeling scheme is specified for the tree structure by selecting a list item from the 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.</p>	<ul style="list-style-type: none"> • 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. 	<ul style="list-style-type: none"> • 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 <code>FND_TS_DATA_SOURCE</code>, or correct the primary keys registered in that table to match the new view object definition.
-------------------------------------	---	---	---

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

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

Specifying Data Sources for Tree Structures: Points to Consider

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

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

Labeling Schemes

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

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

Restriction of Tree Node Values

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

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

Data Source Values and Parameters

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

Note

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

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

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

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

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

Specifying Performance Options for a Tree Structure: Points to Consider

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

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

Row Flattening

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

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

Column Flattening

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

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

Column Flattened Entity Objects

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

ADF Business Component View Objects

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

Manage Tree Labels

Tree Labels: Explained

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

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

Labeling Scheme	Description
Level	Labels that are automatically assigned based on the data source to which the tree node belongs. A level label points to a specific data source. For example, in a tree that reflects the organizational hierarchy of an enterprise, all division nodes appear on one level and all department nodes on another.
Group	Labels that you can arbitrarily assign to tree nodes.
Depth	Labels that are automatically assigned based on the depth of the tree node within the tree. No manual assignment is performed. <hr/> 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.

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

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

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

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

Terminal Node	On the Data Source dialog box, if the 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 table or in an entity that has been established by tree management as the default

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

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

Managing Tree Nodes

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

Node Levels

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

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

Node Types

A tree node has the following node types.

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

Define Profile Options

Profile Options: Explained

Profile options manage configuration data centrally and influence the behavior of applications. Profile options serve as permanent user preferences and application configuration parameters. You configure profile options with

settings for specific contexts or groups of users. Users customize how their user interfaces look and behave by changing the values of available profile options.

Profile options store the following kinds of information.

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

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

Profile Option Definition and Configuration

Application developers add new profile options and configure ones that are not to be updated by other users. Application administrators and implementation consultants configure profile options with profile option values that are implemented as updatable.

Profile option definitions consist of the following.

- Profile option name
- Application and module in the application taxonomy
- Profile option values
- Profile options categories
- Profile option levels
- Profile option level hierarchy

Profile options can appear on any user interface page without indication that a profile option is what is being set.

Profile Option Values

Some profile options have predefined profile option values.

The Manage Profile Option Values task flow allows an administrator to set updatable profile option values at the available levels, including the user level. You can access the Manage Profile Option Values task starting in the Setup and Maintenance Overview page and searching for profile option tasks.

You can set profile option values at different levels: site, product, and user. The following table provides examples.

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

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

Context such as user session or accessed product determines which profile option value is associated with the profile option name. In the example, if manager1 does not set a profile option value for this profile option, access to Financials for EMEA shows currency in Euros; and access to other products shows currency in UK pounds sterling.

Profile Option Categories

Categories group profile options based on their functional area. Profile option categories facilitate searching and defining data security.

For example, in Oracle Fusion Receivables, the Transactions profile option category groups profile options related to setting how Receivables transactions are to be processed, such as Require Adjustment Reason.

A profile option can be in more than one category.

Profile Option Hierarchies and Levels

Application developers specify at which hierarchy level a profile option is enabled. The predefined profile option hierarchy levels are site, product, and user.

The hierarchy levels specified in the profile option definition determine the context in which a profile option value may be set. If the profile option value at a particular level is updatable, an administrator can update the profile option value for that context.

Note

Profile options should only be enabled for context levels that are appropriate for that profile option. For example, a profile option indicating a global configuration setting should not be enabled at the user level, if users cannot choose a different value for that setting.

For security, one level in the hierarchy is designated as a user level. A profile option may be enabled at any or all hierarchy levels. When enabled at all levels, the predefined ordering of profile option hierarchy levels gives precedence to the values that are set at the user level over values set at the product and site levels, and precedence to values set at the product level to values set at the site level. If there is no value for the current user, then the product value applies. If there is no value for the user or product, then the site value applies.

The table shows the predefined profile option hierarchy and ordering.

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

You can configure updatable values for profile options at one or more levels depending on which levels are enabled in the profile option definition. When a profile is set at more than one level, higher levels of specificity override lower levels of specificity.

In the example, if the currency setting for the site is UK pounds sterling, but the Financials division works in the Netherlands using the Euro, a manager in the US can override that product level setting at the user level to use US dollars when accessing Financials applications.

In another example, if a profile option called Printer is set only at the site and product levels. When a user logs on, the Printer profile option assumes the value set at the product level, since it is the highest level setting for the profile.

Tip

Set site-level profile option values before specifying values at any other level. The profile option values specified at the site-level work as defaults until profile option values are specified at the other levels.

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.

Tasks	Role	Considerations
Planning, creating, and editing a new profile option	Applications developer	<p>Since profile options are for permanent settings, do not use profiles options to cache temporary session attributes.</p> <p>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.</p>
Configure values in an existing profile option	Applications developer, application administrator, and implementation consultant	Manage the values for existing profile options.
Create and edit profile option categories	Applications developer, application administrator, and implementation consultant	Manage categories for organizing existing profile options.

Note

Since a profile option enables a behavior in an application user interface or across applications, a value change made by an end user is reflected in the UI page for managing profile option values.

Before Creating a Profile Option

Profile options are best defined for managing configuration data centrally and influencing the behavior of applications.

If the purpose of a profile option setting is specific to a piece of data (typically setup data), it is best implemented as an attribute of that data.

Do not use profile options for behavior that is not configurable.

Profile options exist independent of role.

Do not use profile options to implement function security. For example, an application should not check for a profile option value set to yes to provide access to a page. Do not use profile options to implement data security, such as a profile option value that must be set to a specific value to provide view access to an entity.

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

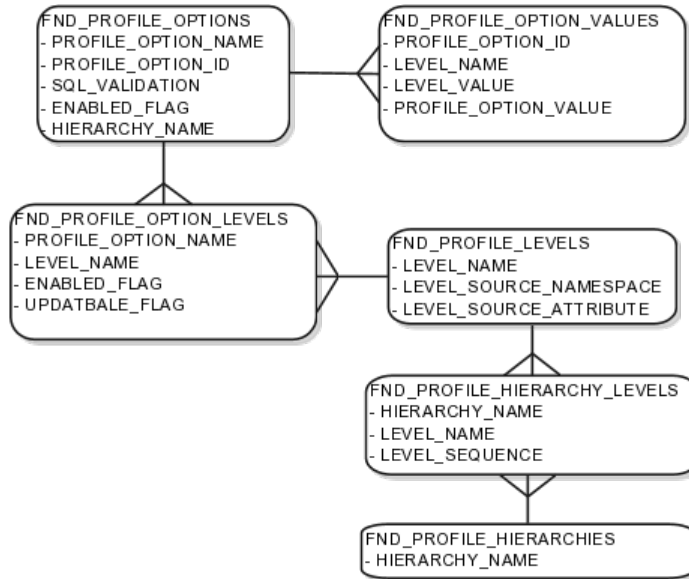
Evaluate if there is a genuine need before creating a profile option. Do not force users to make a decision about an aspect of their application use that is of no concern.

Evaluating need includes looking for duplicate or similar profile options, even in other products, before creating a new one. For example, you do not need multiple profile options to choose a preferred currency.

Profile Options Data Model

The profile option data model illustrates the relationships among profile option elements.

The figure shows the data model of profile option entities.



For more information about planning profile options, see the Oracle Fusion Applications Developer's Guide.

Managing Profile Options: Points to Consider

A profile option definition consists of a name for the profile option and valid values. It is defined within a module of the application taxonomy. Application developers manage profile options to create new profile options or modify existing profile option definitions, which includes specifying the levels at which a profile option is enabled and defining values. Implementation consultants and application administrators configure existing profile options by managing the profile option's updatable values, and creating categories that group profile options.

Configuring a Profile Option

A profile option definition includes information about the owning application and module in the application taxonomy. A start or end date, or both may limit when a profile option is active. The profile option definition may include an SQL validation statement that determines which values are valid, and the hierarchy levels at which the profile option is enabled and updatable.

To be visible to users, a profile option must be user enabled. You can also allow user updates of the profile option, which means users can make changes to the validation and the profile option level information.

Profile option levels specify at which context level profile values may be enabled or updated

Profile options should only be enabled for context levels that are appropriate for that profile option. For example, a profile option indicating a global configuration setting should not be enabled at the user level, if users cannot choose a different value for that setting.

SQL Validation

The SQL validation of the profile option definition determines what valid profile option values are available. In the absence of validation, any value is valid.

For example, SQL validation provides a means of defining a list of values for the valid values of the profile option. The SQL validation can use lookups to provide the valid values for profile options, such as the lookup codes of the YES_NO lookup type.

With a profile option called DEFAULT_LANGUAGE, you can configure the following validation.

```
SELECT DESCRIPTION Language, NLS_LANGUAGE
FROM FND_LANGUAGES_VL
WHERE INSTALLED_FLAG IN ('B','I')
ORDER BY DESCRIPTION
```

This results in the following list of values based on data in FND_LANUGUAGE_VL.

Display Value	Hidden Value
American English	US
French	F
Spanish	E

Hidden values must be varchar2(2000).

Profile options generally provide configuration values within a particular context. Though you can create a profile option to be global, think of global values as default values to avoid storing inappropriate configuration information as profile option values. Create global profile options that have corresponding contextual levels.

Managing Profile Option Categories: Points to Consider

Use profile option categories to group profile options.

Organizing Profile Options in Categories

As a guideline, group profile options in a single category if the profile options affect the same feature, or if an administrator would likely want to see the profile options in the results of a single search.

Application developers are responsible for the initial groupings and then administrators can make changes based on their specific needs. Administrators

can categorize profile options and then easily search on profile options by category.

Tip

Define profile option categories first and assign new profile options to existing categories rather than defining profile options first and then defining categories to categorize them.

Adding New Profile Option Categories

You can add new categories or add profiles to an existing category.

You can create a profile option category by duplicating an existing category and editing it for a new grouping of profile options. You can add multiple profile options to a category. A profile option can exist in multiple categories.

Profile Option Order in a Category

Specify a profile option sequence to determine the order of profile options when queried by profile option category.

Viewing and Editing Profile Option Values: Points to Consider

A profile option value consists of the value and the context or level where the value is set. You specify the context with a pairing of the profile option value's level and level value, such as the product level and the level value GL for Oracle Fusion General Ledger. Adding or modifying profile option values can include deciding which valid values are enabled or updatable at which level.

The SQL validation of the profile option definition determines what valid profile option values are available. In the absence of validation, any value is valid.

Profile Option Levels and User Session Context

Site level profile option values affect the way all applications run for a given implementation. Product level profile option values affect the way applications owned by a particular product code behave. For example, a product may use profile options set at the product level to determine how regions provided by a common module such as those available from Oracle Fusion Trading Community Model or Customer Relationship Management (CRM) display in a particular work area or dashboard. User level profile option values affect the way applications run for a specific application user

Whichever profile option value is most specific to a user session, that is the value at which the profile option is set for the user session.

For example, the predefined FND_LANGUAGE profile option sets the default language. In addition to a site level value, you can define a value for various product or user levels.

Level Name	Level Value	Profile Option Value
Site	InFusion	American English

Product	Customer Center	French
Product	CRM Application Composer	American English
User	Application Administrator	American English
User	Hima	Hindi

Values at the site level take effect for any user unless overridden by a different value set at the more specific levels of product and user. Product level profile option values affect the way applications owned by a particular product code behave. In addition to user level profile option values in applications, selections may be available in the user preferences workspace.

The following table demonstrates the FND_LANGUAGE profile option settings that would apply to specific users, based on the example above. For example, the user Hima is using the CRM Application Composer product, in the InFusion site. The example above shows that this profile option is set to Hindi at the user level for Hima. Because user is the highest applicable level for Hima, the applicable profile option value is Hindi for Hima.

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

Note

More than one site level value is relevant in an enterprise with multiple tenants using a single instance of Oracle Fusion Applications.

Effect of Changes to Profile Option Values

Any change you make to a user level profile option has an immediate effect on the way applications run for that session. When you sign in again, changes made to your user level profile options in a previous session are still in effect. When you change profile option value at the product level and no user level values are set, you see the update immediately, but other users may not see the changed value until signing out and back in. When you change a profile option value and the new value affects other users, the change takes effect only when users sign in the next time.

Changes to site level profile options take effect for any user session that is started after the setting has been changed. Changes to site or user level profile options do not affect any user sessions that are already in progress when the change is made.

Changes to site or user level profile options take effect for any C or PL/SQL processes, such as scheduled jobs, that are launched after the setting has been changed. Profile option changes do not affect C or PL/SQL processes that are already running.

Define Flexfields

Flexfields: Overview

Flexfields are a mechanism for configuring and extending applications to meet enterprise-specific needs by adding an attribute to a business component. Using flexfields, you can extend application objects to capture data that would not otherwise be tracked by the application, or configure intelligent key codes comprised of meaningful parts according to your business practices. Flexfields encapsulate all of the pieces of information related to a specific purpose, such as a key identifying a particular purchase, or the components of a student's contact information, or the features of a product in inventory.

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

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

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

Accessing Flexfields

Manage flexfields using tasks you access by starting in the Setup and Maintenance Overview page. This page is available from the Administrator menu.

Tip

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

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 expands a customizable area of an invoices page by providing subfields for collecting invoice details. You can implement these subfields, which are descriptive flexfield segments, as context sensitive so they appear only when needed. Extensible flexfields are similar to descriptive flexfields, but provide additional advanced features. Generally setup of descriptive and extensible flexfields is optional because their segments capture additional information that is stored attributes on an entity

Segments

All flexfields consist of segments. Segments represent attributes of information. They can appear globally wherever the flexfield is implemented, or based on a structure or context.

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

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

Value Sets

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

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

Structure and Context

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

Each key flexfield structure is a specific configuration of segments. Adding or removing segments, or rearranging their order, produces a different structure.

The database columns on which segments in different structures are based can be reused in as many structures as desired.

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

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

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

Deployment

A flexfield must be deployed to display its current definition in a runtime application user interface. For example, if the deployment status is Edited, the flexfield segments may appear in the UI based on the flexfield definition at the time of last deployment, rather than the current definition.

You can deploy a flexfield as a sandbox for testing the configuration before deploying it to the mainline for all users. In the case of extensible flexfields, you can deploy offline as a background process.

Runtime Appearance

In an application user interface, descriptive flexfield segments appear as label and field pairs or as a table of fields where the column headers correspond to the labels. The fields represent the flexfield segments and accept entered input or a selection from a list of choices that correspond to the segment's assigned value set. Extensible flexfield segments appear grouped within labeled regions, where each grouping is a context and the region labels are the context names.

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

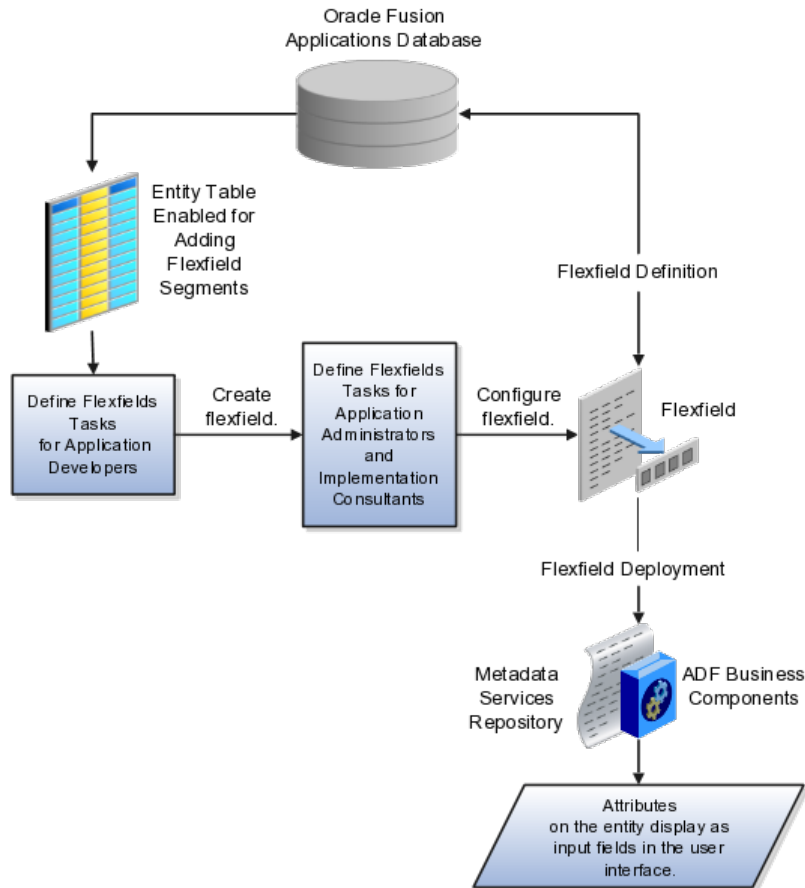
All segments of a single flexfield are grouped together by default. The layout and positions of the flexfield segments depend on where the application developer places the flexfield on the page. Flexfields may also be presented in a separate section of the page, in a table, or on their own page or subwindow.

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

Flexfields and Oracle Fusion Application Architecture: How They Work Together

Administrators configure flexfield segments to capture data that represents the values of attributes. Flexfield segments represent attributes of entities (business objects). Most business objects are enabled for descriptive flexfields. Some business objects are enabled for extensible flexfields.

The figure shows the layers of a flexfield: the business entity table and metadata in the database, business components that are Application Development Framework (ADF) objects or ADFbc objects derived from the metadata and stored in the Metadata Services Repository (MDS), and the user interface where the input fields defined by the flexfield segments are rendered. The flexfield definition consists of all the metadata defined during configuration and stored in the database.



Application developers create a flexfield and register it so that it is available for configuration. Administrators and implementation consultants configure segments and other properties of the available flexfields. This information is stored as additional flexfield metadata in the database. Deploying the flexfield generates ADF business components based on the flexfield metadata in the database.

Integration

A flexfield affects the Web Services Description Language (WSDL) schemas exposed by ADF services and used by SOA composites. The Web services that expose base entity data also expose flexfield segment data.

Attributes incorporate into SOA infrastructure (BPEL, Rules) and integrate with business intelligence (Oracle Business Intelligence, Extended Spread Sheet Database (ESSbase)).

Flexfield configurations are preserved across Oracle Fusion Applications updates.

Deployment

The metadata for the flexfield is stored in the application database as soon as you save your configuration changes. Deploying the flexfield generates the ADF business components so that the runtime user interface reflects the latest definition of the flexfield in the metadata.

Importing and Exporting

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

Runtime

For a flexfield to reflect the latest flexfield definition at runtime it must be deployed. The user interface accesses a business object and the deployed flexfield definition indicates which business object attributes the flexfield captures values for. If you add display customizations for a flexfield using Oracle Composer, these are customizations on the page so that the same flexfield segments can appear differently on various different pages.

Values entered for segments are validated using value sets.

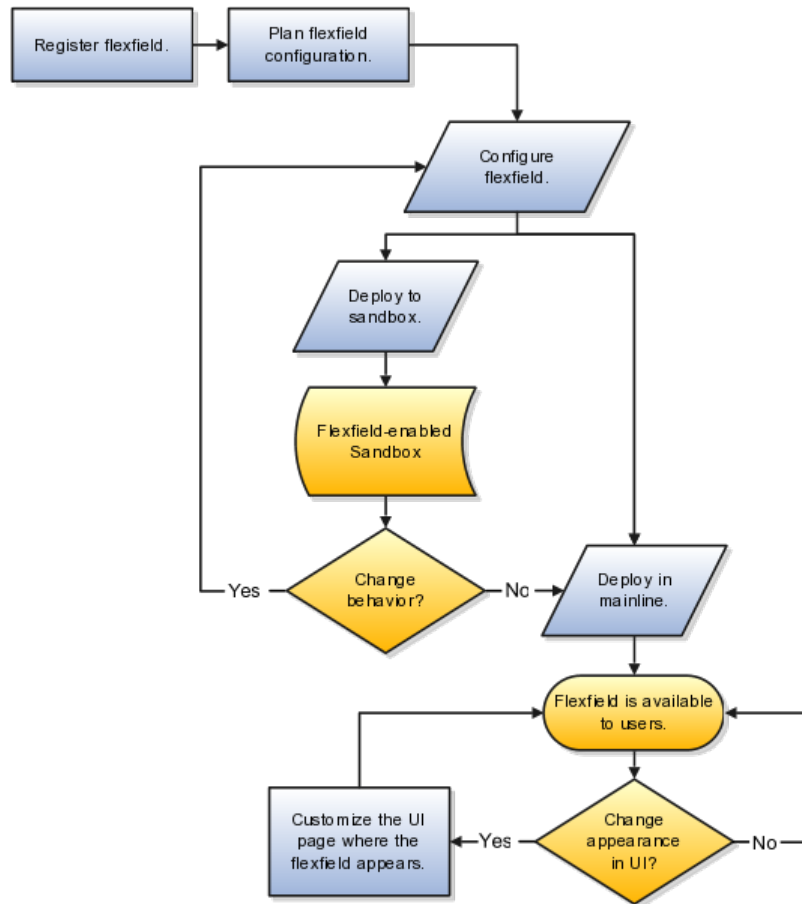
Flexfield Management

Managing Flexfields: Points to Consider

Managing flexfields involves, registering, planning, and configuring flexfields.

You plan and configure the registered flexfields provided in your applications by applications developers. How you configure flexfield segments determines how the flexfield segments appear to end users. Optionally, you can customize the UI page to change how the flexfield segments appear to end users on that page.

The figure shows the processes involved in making flexfields available to end users. The tasks in the Define Flexfields activity allow administrators to configure and deploy flexfields. If you deploy a flexfield to a sandbox and decide not to make further changes, you select the flexfield in the Manage Flexfields tasks of the Define Flexfields activity and deploy the flexfield in the mainline so it is available to users.



Registering Flexfields

Application development registers flexfields so they are available to administrators and implementation consultants for configuration.

Application development creates the capacity of database tables to support flexfields so an enterprise can capture specific information about an entity. Many flexfields are predefined in Oracle Fusion Applications.

A flexfield must be registered before it can be configured.

For more information on registering flexfields, see Oracle Fusion Applications: Developer's Guide.

Planning Flexfields

All flexfields consist of segments which represent attributes of an entity. The values an end user inputs for an attribute are stored in a column of the entity table. Application development enables columns of entity tables for use in flexfields during flexfield registration.

A flexfield must be registered before it can be configured. Before configuring new flexfield segments for your enterprise, be sure to plan their implementation carefully.

For more information on planning flexfield configuration, see Oracle Fusion Applications Extensibility Guide.

Configuring Flexfields

Administrators or implementors configure flexfields so they meet the needs of the enterprise. Some flexfields require configuration to make an application operate correctly.

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 runtime page.
 - Use the edit icon to manage a flexfield.
 - Use the quick create icon to add descriptive 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 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.

Some descriptive flexfields provide parameters that are attributes of the same or related entity objects. You use parameters to set the initial value or derivation value of an attribute from external reference data, such as a column value or a session variable, rather than from user input.

Enabling a Flexfield Segment for Business Intelligence

You can enable key flexfield segment instances and descriptive flexfield segments for business intelligence if the flexfield is registered in the database as an Oracle Business Intelligence-enabled flexfield. For more information on enabling segments for business intelligence, see points to consider when enabling key and descriptive flexfield segments for business intelligence.

Deploying Flexfields

After you configure or change a flexfield, you must deploy it to make the latest definition available to end users.

You deploy a flexfield in the mainline for general use in a test or production environment, or you can deploy a flexfield as a flexfield-enabled sandbox to confirm that it is correctly configured before deploying it to the mainline.

Deploying a flexfield results in a deployment status. Once deployed, the deployment status indicates the state of the currently configured flexfield relative to the last deployed definition

Optionally Changing How Flexfield Segments Appear in a User Interface Page

The flexfield attributes you define integrate with the user interface pages where users access the attributes' business object. Application development determines the UI pages where business objects appear and the display patterns used by default to render flexfield segments.

You can customize the appearance of the flexfield segments in the UI page using Oracle Composer once the flexfield is deployed to the mainline.

Identifying Flexfields on a Runtime Page

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

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

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

Flexfield Segment Properties: Explained

Independent of the value set assigned to a segment, segments may have properties such as validation.

Range Validation of Segments

Range validation enables you to enforce an arithmetic inequality between two segments of a flexfield. For example, a product must be ordered before it can be shipped, so the order date must be on or before the ship date, and consequently the order date segment value must be less than or equal to the ship date segment value. You can use range validation to ensure this relationship.

The conditions for range validation are as follows.

- Segments must be configured for range validation in pairs, one with the low value and one with the high value.
- Both segments must be the same data type.
- Both segments must be part of the same structure in a key flexfield or part of the same context in a descriptive flexfield or extensible flexfield.
- The low value segment must have a lower sequence number than the high value segment.
- Non-range validated segments can exist between a range validated pair, but range validated pairs cannot overlap or be nested.

You can configure as many range validated pairs as you want within the same flexfield. Your application automatically detects and applies range validation to the segment pairs that you define, in sequence order. It must encounter a low value segment first, and the next range validated segment it encounters must be a high value segment. These two segments are assumed to be a matching pair. The low value and the high value can be equal.

Code Naming Conventions

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

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

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

Flexfields and Value Sets: How They Work Together

Value sets are specific to your enterprise. When gathering information using flexfields, your enterprise's value sets validate the values your users enter based on how you defined the value set.

You can assign a value set to any number of flexfield segments in the same or different flexfields. Value set usage information indicates which flexfields use the value set.

Defining and Assigning Value Sets

As a flexfield guideline, define value sets before configuring the flexfield, because you assign value sets to each segment as you configure a flexfield.

Caution

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

Shared Value Sets

When you change a value in a shared value set, the change affects the value set for all flexfields that use that value set. The advantage of a shared values set is that a single change propagates to all usages. The drawback is that the change shared across usages may not be appropriate in every case.

Deployment

When you deploy a flexfield, the value sets assigned to the segments of the flexfield provide end users with the valid values for the attributes represented by the segments.

Defaulting and Deriving Segment Values: Explained

To populate a flexfield segment with a default value when a row is created, specify a default type of constant or parameter and a default value.

To synchronize a segment's value with another field's value whenever it changes, specify the derivation value to be the flexfield parameter from which to derive the attribute's value. Whenever the parameter value changes, the attribute's value is changed to match. If you derive an attribute from a parameter, consider making the attribute read-only, as values entered by users are lost whenever the parameter value changes.

When defaulting or deriving a default value from a parameter, only those attributes designated by development as parameters are available to be chosen.

Different combinations of making the segments read only or editable in combination with the default or derivation value or both, have different effects.

Initial runtime behavior corresponds to the row for the attribute value being created in the entity table. If the default value is read only, it cannot subsequently be changed through the user interface. If the default value is not read only, users can modify it. However, if the segment value is a derived value, a user-modified segment value is overwritten when the derivation value changes.

Default Type	Default value specified?	Derivation value specified?	Initial runtime behavior	Runtime behavior after parameter changes
None	No	Yes	No initial segment value	Changed parameter derivation value updates segment value
Constant	Yes	No	Default segment value	N/A
Constant	Yes	Yes	Default segment value	Changed parameter derivation value updates segment value
Parameter	Yes	No	Default segment value is the parameter's default value	N/A
Parameter	Yes	Yes, and same as default value	Default segment value is parameter's default and derivation value	Changed parameter derivation value updates segment value
Parameter	Yes	Yes, and different from default value	Default segment value is parameter's default value	Changed parameter default value does not update segment value. Only the changed derivation value updates the segment value.

Flexfield Usages: Explained

Usage affects various aspects of flexfields. The usage of the flexfield is set when the flexfield is registered and specifies the application and table with which the flexfield is associated.

Entity usage indicates the table containing the segments of a flexfield.

A flexfield can have multiple usages. The first table registered for a flexfield is the master usage. Segments are based on the master usage, and other usages of the same table for the same flexfield use the same segment setup, though the column names optionally may have a differentiating prefix.

Extensible Flexfields

You can configure different behavior for extensible flexfield contexts at the usage level. The usage of an extensible flexfield context determines in which scenarios or user interfaces the segments of a context appear to end users. For example, if a Supplier page displays an extensible flexfield's supplier usage and a buyer page displays that same extensible flexfield's buyer usage, a context that is associated to the supplier usage but not the buyer usage displays only on the supplier page and not the buyer page.

Value Sets

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

Flexfield Deployment

Flexfield Deployment: Explained

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

After you configure or change a flexfield, you must deploy it to make the latest definition available to end users. You can deploy a flexfield to a sandbox for testing or to the mainline for use. In the case of extensible flexfields, you can deploy offline as a background process.

Deployment Status

Every flexfield has a deployment status.

A flexfield can have the following deployment statuses.

Deployment Status	Meaning
Edited	The flexfield metadata definition has not been deployed yet. Updates of the metadata definition are not applied in the runtime environment yet.

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

Note

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

Initial Deployment Status of Flexfields

The Oracle Fusion Applications installation loads flexfield metadata into the database. This initial load sets the flexfield status to Edited to indicate that the flexfield has not been deployed yet. The application provisioning process during installation deploys the predefined flexfields of the provisioned applications, which sets their status to Deployed if no errors are encountered.

When accessing a provisioned application, deployed flexfields are ready to use. In some cases, flexfield availability at runtime requires setup, such as defining key flexfields.

Metadata Validation

Use the Validate Metadata command to view possible metadata errors before attempting to deploy the flexfield. Metadata validation is the initial phase of 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. Errors in the metadata validation phase of deployment cause the deployment attempt to abort. Metadata validation results do not affect the deployment status of a flexfield.

Flexfield Deployment Status: How It Is Calculated

Flexfield deployment status indicates how the flexfield metadata definition in the Oracle Fusion Applications database relates to the Application Development

Framework (ADF) business components generated into a Metadata Services (MDS) repository.

Settings That Affect Flexfield Deployment Status

If you have made a change to a flexfield and expect a changed deployment status, be sure you have saved your changes. No settings affect flexfield deployment status.

How Flexfield Deployment Status Is Calculated

If the flexfield definition has been edited through the Define Flexfields activity task flows, the status is Edited. The latest flexfield metadata definition in the Oracle Fusion application diverges from the latest deployed flexfield definition. Any change, including if a value set used in a flexfield changes, changes the deployment status to Edited. If a flexfield has never been deployed, its status is Edited.

Note

When an application is provisioned, the provisioning framework attempts to deploy all flexfields in that application.

If you deploy the flexfield to a sandbox successfully, the status is Deployed to Sandbox. The latest flexfield metadata definition in the Oracle Fusion application matches the metadata definition that generated ADF business components in a sandbox MDS repository. Whether the sandbox is active or not does not affect the deployment status. If the flexfield was deployed to a sandbox and has not been edited or re-deployed to the mainline since then, the status remains Deployed to Sandbox independent of whether the sandbox is active, or who is viewing the status.

If you deploy the flexfield successfully, meaning to the mainline, the status is Deployed. The latest flexfield metadata definition in the Oracle Fusion application matches the metadata definition that generated ADF business components in a mainline MDS repository. Change notifications are sent when a flexfield is deployed successfully to the mainline.

If either type of deployment fails so the current flexfield definition is not deployed, the status is Error. The deployment error message gives details about the error. The latest flexfield metadata definition in the Oracle Fusion application likely diverges from the latest successfully deployed flexfield definition.

If the flexfield definition has been modified by a patch, the status is Patched. The latest flexfield metadata definition in the Oracle Fusion application diverges from the latest deployed flexfield definition. If the flexfield definition was Deployed before the patch and then a patch was applied, the status changes to Patched. If the flexfield definition was Edited before the patch and then a patch was applied, the status will remain at Edited to reflect that there are still changes (outside of the patch) that are not yet in effect.

When a deployment attempt fails and you can access the Deployment Error Message for details.

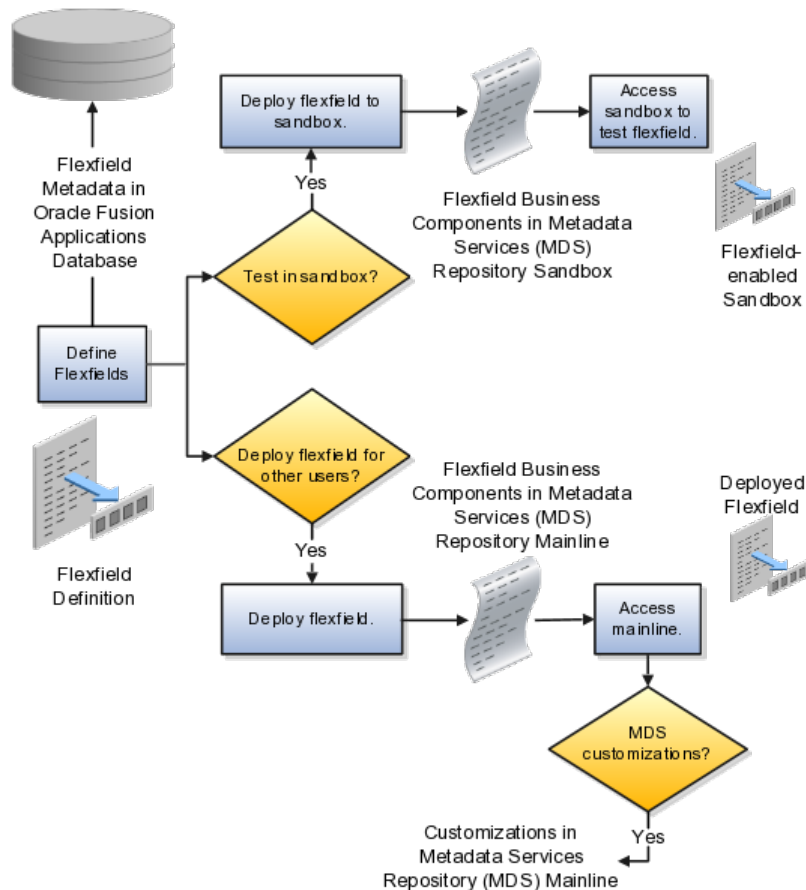
Deploying a Flexfield-Enabled Sandbox: How It Works With Mainline Metadata

The flexfield definition in a sandbox corresponds to the flexfield metadata definition in the Oracle Fusion Applications database at the time the flexfield was deployed to the sandbox. When the flexfield is ready for end users, the flexfield must be deployed to the mainline.

A flexfield-enabled sandbox uses the following components.

- Flexfield metadata in the Oracle Fusion Applications database
- Flexfield business components in a sandbox Metadata Services (MDS) repository
- User interface customizations for the flexfield in the mainline MDS repository

The figure shows the two types of deployment available in the Manage Flexfields activity. Deploying a flexfield to a sandbox creates a sandbox MDS repository for the sole purpose of testing flexfield behavior. The sandbox is only accessible to the administrator who activates and accesses it, not to users generally. Deploying a flexfield to the mainline applies the flexfield definition to the mainline MDS repository where it is available to end users. After deploying the flexfield to the mainline, customize the page where the flexfield segments appear. Customization of the page in the sandbox MDS repository cannot be published to the mainline MDS repository.



Sandbox Metadata Services Repository Data

Deploying the flexfield to a sandbox generates the Application Development Framework (ADF) business components of a flexfield in a sandbox MDS repository for testing in isolation.

Warning

Do not make changes to flexfield segment display features in a flexfield-enabled sandbox as these changes will be lost when deploying the flexfield to the mainline.

Mainline Metadata Services Repository Data

The Oracle Fusion Applications database stores the single source of truth about a flexfield. From this the ADF business component objects that implement the flexfield in the runtime user interface are generated in the mainline MDS repository when the flexfield is deployed.

Deploying a Flexfield-Enabled Sandbox: Points to Consider

Deploying a flexfield to a sandbox creates a flexfield-enabled sandbox . Each flexfield-enabled sandbox contains only one flexfield.

You can test the runtime behavior of a flexfield in the flexfield-enabled sandbox. If changes are needed, return to the Define Flexfield tasks to change the flexfield definition.

When you deploy a flexfield to sandbox, the process reads the metadata about the segments from the database, generates flexfield Application Development Framework (ADF) business component artifacts based on that definition, and stores in the sandbox only the generated artifacts derived from the definition.

Sandbox MDS Repository Data

The sandbox data allows you to test the flexfield in isolation without first deploying it in the mainline where it could be accessed by users.

Warning

Do not make changes to flexfield segment display features in a flexfield-enabled sandbox as these changes will be lost when deploying the flexfield to the mainline.

Managing a Flexfield-Enabled Sandbox

When you deploy a flexfield as a sandbox, that flexfield-enabled sandbox automatically gets activated in your user session. When you sign back in to see the changes, the sandbox is active in your session.

You can only deploy a flexfield to a sandbox using the Define Flexfields task flow pages.

You also can use the Manage Sandboxes feature in the Administration menu of the Setup and Maintenance work area to activate, access, or delete a flexfield-enabled sandbox.

Note

Whether you use the Define Flexfields or Manage Sandboxes task flows to access a flexfield-enabled sandbox, you must sign out and sign back in before you can see the changes you deployed in the runtime.

You cannot publish the flexfield from the sandbox to the mainline. You must use the Define Flexfields task flow pages to deploy the flexfield for access by users of the mainline because the flexfield configuration in the mainline is the single source of truth.

Deploying Flexfields Using the Command Line: Explained

You can use the Manage Key Flexfields, Manage Descriptive Flexfields, and Manage Extensible Flexfields tasks to deploy flexfields. You can also use WebLogic Server Tool (WLST) commands for priming the Metadata Services (MDS) repository with predefined flexfield artifacts and for deploying flexfields.

The table describes the available commands.

WebLogic Server Tool Command	Description
<code>deployFlexForApp</code>	Deploys all flexfields for the specified enterprise application. Only flexfields whose status is other than deployed are affected by this command unless the option is enabled to force all flexfields to be deployed regardless of deployment status. Initial application provisioning runs this command to prime the MDS repository with flexfield artifacts.
<code>deployFlex</code>	Deploy a single flexfield regardless of deployment status
<code>deployPatchedFlex</code>	Deploys flexfield changes that have been delivered using a flexfield Seed Data Framework (SDF) patch. Deploys flexfields that have a Patched deployment status.
<code>deleteFlexPatchingLabels</code>	Displays MDS label of flexfield changes for viewing and deleting patching labels.

Executing these commands outputs a report at the command line. The report provides the following information for every flexfield that is processed.

- Application identity (APPID)
- Flexfield code
- Deployment result, such as success or error

In case of errors, the report lists the usages for which the errors were encountered. If a runtime exception occurs, the output displays the traceback

information. For each WLST flexfield command, adding the `reportFormat='xml'` argument returns the report as an XML string.

Consider the following aspects of command line deployment.

- Preparing to use the WLST flexfield commands
- Using the `deployFlexForApp` command
- Using the `deployFlex` command
- Using the `deployPatchedFlex` command
- Using the `deleteFlexPatchingLabels` command
- Exiting the WLST and checking the results

Preparing To Use the WLST Flexfield Commands

You can only execute the WLST flexfield commands on a WebLogic Administration Server for a domain that has a running instance of the Oracle Fusion Middleware Extensions for Applications (Applications Core) Setup application.

For more information on deploying the Applications Core Setup application, see the Oracle Fusion Applications Developer's Guide.

Ensure that the AppMasterDB data source is registered as a JDBC data source with the WebLogic Administration Server and points to the same database as the ApplicationDB data source.

Start the WebLogic Server Tool (WLST) tool, if it is not currently running.

UNIX:

```
sh $JDEV_HOME/oracle_common/common/bin/wlst.sh
```

Windows:

```
wlst.cmd
```

Connect to the server, replacing the user name and password arguments with your WebLogic Server user name and password.

```
connect('wls_username', 'wls_password', 'wls_uri')
```

The values must be wrapped in single-quotes. The `wls_uri` value is typically `T3://localhost:7101`.

For more information on the WLST scripting tool, see the Oracle Fusion Middleware Oracle WebLogic Scripting Tool.

Using the `deployFlexForApp` Command

The `deployFlexForApp` command translates the product application's predefined flexfield metadata into artifacts in the MDS repository.

Important

This command is run automatically when you provision applications. However, after custom applications development, you must run the `deployFlexForApp`

command after you configure your application to read the flexfield artifacts from the MDS repository and before you log into the application for the first time, even if there is no predefined flexfield metadata.

This command does not deploy flexfields that have a status of Deployed unless the force parameter is set to 'true' (the default setting is 'false').

For more information on priming the MDS partition with configured flexfield artifacts, see the Oracle Fusion Applications Developer's Guide.

From the WLST tool, execute the following commands to deploy the artifacts to the MDS partition, replacing `product_application_shortname` with the application's short name wrapped in single-quotes.

```
deployFlexForApp('product_application_shortname'[, 'enterprise_id']  
[, 'force'])
```

In a multi-tenant environment, replace `enterprise_id` with the Enterprise ID to which the flexfield is mapped. Otherwise, replace with 'None' or do not provide a second argument.

To deploy all flexfields regardless of their deployment status, set force to 'true' (the default setting is 'false'). If you want to deploy all flexfields in a single-tenant environment, you either can set `enterprise_id` to 'None', or you can use the following signature:

```
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.

```
deployFlex('flex_code', 'flex_type')
```

The values must be wrapped in single-quotes.

Using the `deployPatchedFlex` Command

Use the `deployPatchedFlex` command for situations where the patching framework does not invoke the command, such as when an application has been patched offline.

If the installation is multi-tenant enabled, the command deploys all patched flexfields for all enterprises. This command is not intended to be invoked manually.

Check with your provisioning or patching team, or the task flows for managing flexfields, to verify that the flexfield has a Patched deployment status.

From the WLST tool, execute the following command to deploy the artifacts to the MDS partition of all flexfields that have a READY status.

```
deployPatchedFlex()
```

Execute the following command to deploy all flexfields that have either a READY status or an ERROR status.

```
deployPatchedFlex(mode='RETRY')
```

Using the deleteFlexPatchingLabels Command

Whenever you deploy flexfield changes to MDS using the `deployPatchedFlex()` WLST command, an MDS label is created in the format `FlexPatchingWatermarkdate+time`. Use the `deleteFlexPatchingLabels` command to inquire about and delete these labels.

From the WLST tool, execute the `deployPatchedFlex()` command with no arguments to delete the flexfield patching labels.

To output a list of flexfield patching labels, execute the command with the `infoOnly` argument, as follows:

```
deleteFlexPatchingLabels(infoOnly='true')
```

Exiting the WLST and Checking the Results

To exit the tool, execute the following command.

```
disconnect()
```

Optionally, sign into the application, access user interface pages that contain flexfields, and confirm the presence of flexfields for which configuration exists, such as value sets, segments, context, or structures.

Manage Value Sets

Value Sets: Explained

A value set is a set of valid values that you assign to a flexfield segment.

An end user enters a value into a flexfield segment while using the application. The flexfield validates the segment against the set of valid values that you configured as a value set and assigned to the segment.

For example, you can define a required format, such as a five digit number, or a list of valid values, such as green, red, and blue.

Flexfield segments are usually validated, and typically each segment in a given flexfield uses a different value set. You can assign a single value set to more than one segment, and you can share value sets among different flexfields.

Caution

Be sure changes to a shared value set are compatible with all flexfields segments using the value set.

Defining value sets involves making decisions about the following.

- Validation
- Security
- Precision and scale
- Usage and deployment

Validation

The following types of validation are available for value sets.

- Format only, where end users enter data rather than selecting values from a list
- Independent, a list of values consisting of valid values you specify
- Dependent, a list of values where a valid value derives from the independent value of another segment
- Subset, where the list of values is a subset of the values in an existing independent value set
- Table, where the values derive from a column in an application table and the list of values is limited by a WHERE clause

A segment that uses a format only value set does not present a list of valid values to users.

You can build a tree structure from the values in an independent value set whose data type is character.

Note

Adding table validated value sets to the list of available value sets available for configuration is considered a custom task.

For more information, see the Oracle Fusion Applications Extensibility Guide.

Security

Value set security only works in conjunction with usage within flexfield segments. If a value set is used standalone, meaning outside a flexfield, value set security is not applied, but Oracle Fusion data security is enforced.

You can specify that data security be applied to the values in flexfield segments that use a value set. Based on the roles provisioned to users, data security policies determine which values of the flexfield segment end users can view or modify.

Value set security applies at the value set level. If a value set is secured, every usage of it in any flexfield is secured. It is not possible to disable security for individual usages of the same value set.

Value set security applies to independent, dependent, or table-validated value sets.

Value set security applies mainly when data is being created or updated, and to key flexfield combinations tables for query purposes. Value set security does not determine which descriptive flexfield data is shown upon querying.

Security conditions defined on value sets always use table aliases. When filters are used, table aliases are always used by default. When predicates are defined for data security conditions, make sure that the predicates also use table aliases.

For key flexfields, the attributes in the view object that correspond to the code combination ID (CCID), structure instance number (SIN), and data set number (DSN) cannot be transient. They must exist in the database table. For key flexfields, the SIN segment is the discriminator attribute, and the CCID segment is the common attribute.

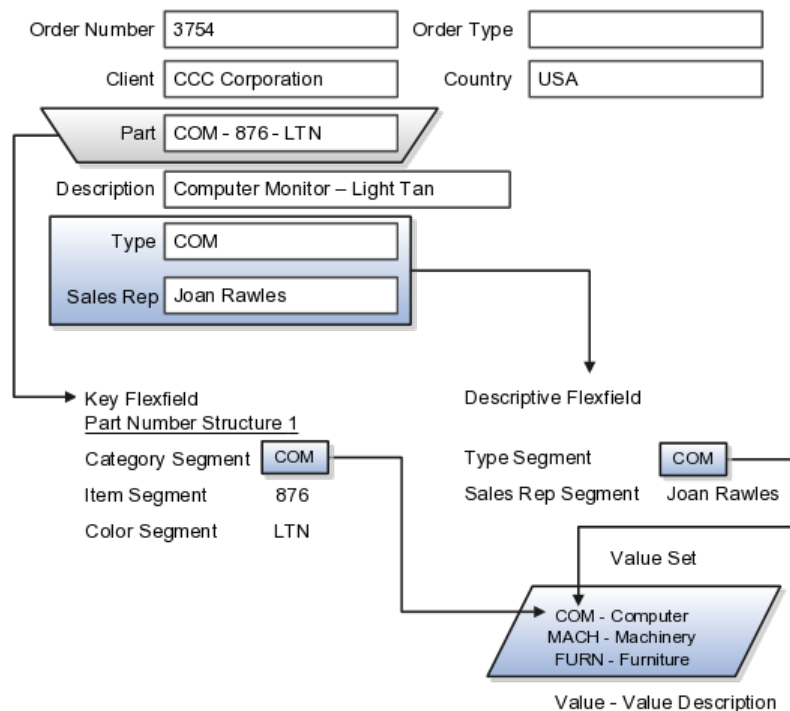
Precision and Scale

For a value set with the data type Number, you can specify the precision (maximum number of digits user can enter) or scale (maximum number of digits following the decimal point).

Usage and Deployment

The usage of a value set is the flexfields where that value set is used. The deployment status of flexfields in which the value set is used indicates the deployment status of the value set instance.

The figure shows a value set used by a segment in a key flexfield and the context segment of a descriptive flexfield.



For most value sets, when you enter values into a flexfield segment, you can enter only values that already exist in the value set assigned to that segment.

Global and context-sensitive segment require a value set. You can assign a value set to a descriptive flexfield context segment. If you specify only context values, not value sets for contexts, the set of valid values is equal to the set of context values.

Defining Value Sets: Critical Choices

Validation and usage of value sets determine where and how end users access valid values for attributes represented by flexfield segments.

Tip

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.

Value Sets for Context Segments

When assigning a value set to a context segment, you can only use table-validated or independent value sets. The data type must be character and the maximum length of the values being stored must not be larger than column length of the context.

Format Only Validation

The format only validation type enables end users to enter any value, as long as it meets your specified formatting rules. That is, the value must not exceed the maximum length you define for your value set, and it must meet any format requirements for that value set.

For example, if the value set allows only numeric characters, your user could enter the value 456 (for a value set with maximum length of three or more), but could not enter the value ABC. A format only value set does not otherwise restrict the range of different values that users can enter. For numeric values, you can also specify if a numeric value should be zero filled or how many digits should follow the radix separator.

Interdependent Value Sets

You cannot specify a dependent value set for a given segment without having first defined an independent value set that you apply to another segment in the same flexfield. You use a dependent value set to limit the list of values for a given segment based on the value that the end user has chosen for a related independent segment. The available values in a dependent list and the meaning of a given value depend on which value was selected for the independently validated segment.

For example, you could define an independent value set of U.S. states with values such as CA, NY, and so on. Then you define a dependent value set of U.S. cities, with values such as San Francisco and Los Angeles that are valid for the independent value CA, and New York City and Albany that are valid for the independent value NY. In the UI, only the valid cities can be selected for a given state.

Because you define a subset value set from an existing independent value set, you must define the independent value set first. End users do not need to choose a value for another segment first to have access to the subset value set.

Table Validation

Typically, you use a table-validated set when the values you want to use are already maintained in an application table (for example, a table of vendor names). Table validation allows you to enable a segment to depend upon multiple prior segments in the same context or structure.

Table-validated value sets have unique values across the table, irrespective of bind variables. The WHERE clause fragment of the value set is considered if it does not have bind variables. If it has bind variables, the assumption is that the values are unique in the value set.

Range

In the case of format, independent, or dependent value sets, you can specify a range to further limit which values are valid. You can specify a range of values that are valid within a value set. You can also specify a range validated pair of segments where one segment represents the low end of the range and another segment represents the high end of the range

For example, you might specify a range for a format-only value set with format type Number where the user can enter only values between 0 and 100. If you use a table value set, you cannot reference flexfield segments in the WHERE clause of the value set. For example, the WHERE clause cannot reference a segment or a value set.

Security

In the case of independent and dependent values, you can specify that data security be applied to the values in segments that use a value set. Based on the roles provisioned to users, data security policies determine which values of the flexfield segment end users can view or modify.

When you enable security on a table-validated value sets, the security rule that is defined is absolute and not contingent upon the bind variables (if any) that may be used by the WHERE clause of the value set. For example, suppose a table-validated value set has a bind variable to further filter the value list to x, y and z from a list of x, y, z, xx, yy, zz. The data security rule or filter written against the value set should not assume anything about the bind variables; it must assume the whole list of values is available and write the rule, for example, to allow x, or to allow y and z. By default in data security all values are denied, and show only rows to which access has been provided.

Maintenance

There is no need to define or maintain values for a table-validated or subset value set, as the values are managed as part of the referenced table or independent value set, respectively.

If your application has more than one language installed, or there is any possibility that you might install one or more additional languages for your application in the future, select **Translatable**. This does not require you to provide translated values now, but you cannot change this option if you decide to provide them later.

For more information about defining value sets, see the Oracle Fusion Applications Extensibility Guide.

Manage Descriptive Flexfields

Descriptive Flexfields: Explained

Descriptive flexfields provide a way to add custom attributes to entities, and define validation and display properties for them. These attributes are generally standalone. They don't necessarily have anything to do with each other and are not treated together as a combination.

All Oracle Fusion Applications business entities that you can access are enabled for descriptive flexfields. Descriptive flexfields are optional. You can choose to configure and expose segments for the descriptive flexfield defined and registered in your database, or not. 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. The segments of a descriptive flexfield are made available to end users as individual fields in the application user interface.

Context

A descriptive flexfield can have only one context segment to provide context sensitivity.

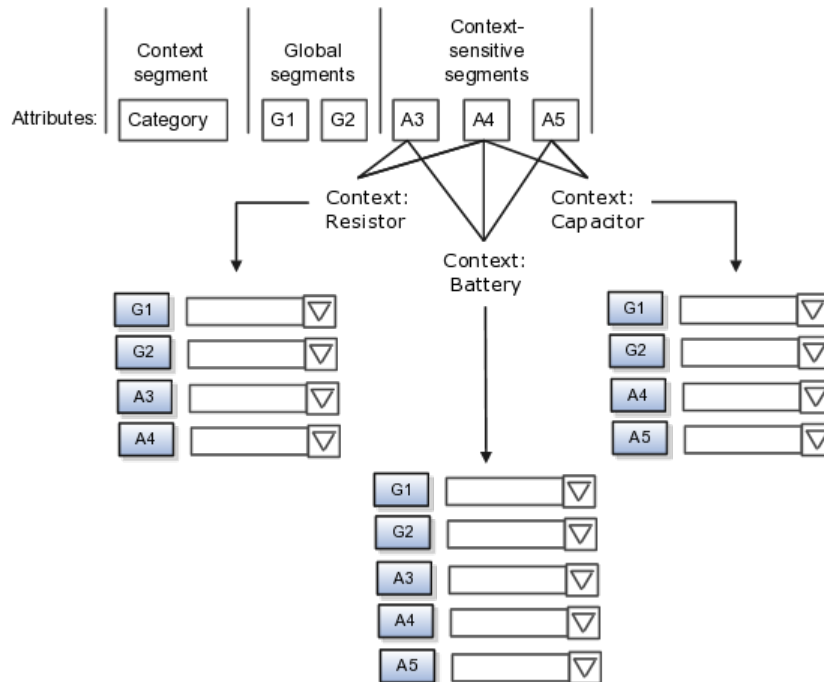
The same underlying column can be used by different segments in different contexts. For example, you can define a Dimensions context that uses the ATTRIBUTE1 column for height, the ATTRIBUTE2 column for width, and the ATTRIBUTE3 column for depth. You can also define a Measurements context that uses the same columns for other attributes: the ATTRIBUTE1 column for weight, the ATTRIBUTE2 column for volume, and the ATTRIBUTE3 column for density.

Segments and Contexts

Descriptive flexfield segments are of the following types.

Segment Type	Runtime Behavior
Global segment	Always available
Context segment	Determines which context-sensitive segments are displayed
Context-sensitive segment	Displayed depending on the value of the context segment

In the figure, a descriptive flexfield has one context segment called Category for which there are three values: Resistor, Battery, and Capacitor. In addition, the descriptive flexfield consists of two global segments that appear in each of the contexts, and three context-sensitive segments that only appear in the context in which they are configured.



Application development determines the number of segments available for configuring. During implementation you determine which attributes to add using the available segments, and the context values and the combination of attributes in each context. A segment can be used for different attributes, such as Height in Context1 and Color in Context2. Each segment of a descriptive flexfield that you make available to end users is exposed in the user interface as an individual field.

Value Sets

For each global and context-sensitive segment, you configure the values allowed for the segment and how the values that end users enter are validated, including interdependent validation among the segments.

Managing Descriptive Flexfields: Points to Consider

Configuring descriptive flexfields involves managing the available flexfields registered with your Oracle Fusion Applications database and configuring their flexfield-level properties, defining and managing descriptive flexfield contexts, and configuring global and context-sensitive segments.

Every descriptive flexfield is registered to include a context segment, which you may choose to use or not.

Segments

You can assign sequence order numbers to global segments and to context-sensitive segments in each context. Segment display is always in a fixed order. You cannot enter a number for one segment that is already in use for a different segment.

Value sets are optional for context segments. The value set that you specify for a context segment consists of a set of context codes, each of which corresponds to a context that is appropriate for the descriptive flexfield. If you do not specify a value set for a context segment, the valid values for that context segment are derived from the context codes. The definition of each context segment specifies the set of context-sensitive segments that can be presented when that context code is selected by the end user.

For reasons of data integrity, you cannot delete an existing context. Instead, you can disable the associated context value in its own value set by setting its end date to a date in the past.

You can configure the individual global segments and context-sensitive segments in a descriptive flexfield. These segment types are differentiated by their usage, but they are configured on application pages that use most of the same properties.

Usages

Descriptive flexfield usages allow for the same definition to be applied to multiple entities. Descriptive flexfield tables define the placeholder entity where the flexfield segment values are stored once you have configured the descriptive flexfield.

Parameters

Parameters are public arguments to a descriptive flexfield. Parameters provide outside values in descriptive flexfield validation. Parameters can be referenced by the logic that derives the default segment value and in table-validated VALUE set WHERE clauses.

Delimiters

A segment delimiter or separator visually separates segment values when the flexfield is displayed as a string of concatenated segments.

Enabling Descriptive Flexfield Segments for Business Intelligence: Points to Consider

A descriptive flexfield that is registered in the database as enabled for Oracle Business Intelligence (BI) includes a BI Enabled setting for each of its segments. When a global, context, or context-sensitive segment is BI-enabled, it is available for use in Oracle Business Intelligence.

The following aspects are important in understanding BI-enabled flexfield segments.

- Flattening business components to use BI-enabled segments in Oracle BI.
- Equalizing context-sensitive segments to prevent duplication and complexity in the flattened components
- Mapping attributes of flattened business components to logical objects in Oracle BI.

- Managing the labels that map segments to logical objects in Oracle BI.

After you deploy a business intelligence-enabled flexfield, import the flexfield changes into the Oracle Business Intelligence repository to make use of the newly-generated attributes in business intelligence applications. For additional information about import, refer to the Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition).

Flattening

When you deploy a business intelligence-enabled descriptive flexfield, the deployment process generates an additional set of flattened business components for use in Oracle BI. The flattened business components include attributes for business intelligence-enabled segments only.

Flattened components include one attribute for the BI-enabled context-segment, and one attribute for each business intelligence-enabled global segment. For BI-enabled context-sensitive segments, if you assigned a label to the segment, the flattened components include an additional single attribute representing segments with that label. If you did not assign a label, the flattened components include a discrete attribute for each BI-enabled context-sensitive segment in each context.

Equalization

You can prevent duplication and the extra workload and complexity that result from the flattening process by assigning a label to any set of context-sensitive segments that serve the same purpose in different contexts. BI-enabled context-sensitive segments that have labels are equalized so that the flattened business components include one attribute for each label, consolidated across contexts. Non-labeled context-sensitive segments are not equalized across context values, so the flattened components include a separate attribute for each context-sensitive segment for each context value.

Note

It may not be possible to equalize similarly labeled segments if they have incompatible data types or value set types.

Mapping to Logical Objects in Business Intelligence

Assign a label to a global segment, context segment, or context-sensitive segment to map the corresponding attribute in the flattened components to a logical object in Oracle Business Intelligence. Using labels to map segments to BI logical objects minimizes the steps for importing the flexfield into Oracle Business Intelligence.

Note

Assigning a label to a context-sensitive segment serves to equalize the attribute across contexts, as well as map the equalized attribute to business intelligence.

Managing Labels

You may assign a predefined label (if available) to segments or create new labels for assignment, as needed. Specify a code, name, and description to identify

each label. In the BI Object Name field, enter the name of the logical object in Oracle Business Intelligence to which the segment label should map during import. Specifying the BI logical object minimizes the steps for importing the flexfield into Oracle Business Intelligence and helps to equalize context-sensitive segments across contexts.

If no labels are assigned to a BI-enabled segment, or the BI Object Name on the assigned label does not exist in business intelligence, you must manually map the segment to the desired logical object when importing into Oracle Business Intelligence.

In addition, context-sensitive segments without labels cannot be equalized across context values. The flattened components include a separate attribute for each non-labeled context-sensitive segment in each context.

Manage Extensible Flexfields

Extensible Flexfields: Explained

Extensible flexfields are like descriptive flexfields, with some additional features.

- You can add as many context-sensitive segments to the flexfield as you need. You are not dependent on the number of segments predefined and registered for the flexfield.
- You can configure a one-to-many relationship between the entity and its extended attribute rows.
 - A row of data can have multiple contexts associated with it.
 - A row of data can have multiple occurrences of the same context.
- You can configure contexts in groups so the attributes in the context always appear together in the user interface.
- You can use existing hierarchical categories so that entities inherit the contexts that are configured for their parents. Contexts are reusable throughout categories.
- You can specify view and edit privileges for the extensible flexfield segments to control who sees the attributes and who can change the attribute's values.

When you configure a context for multiple rows per entity, the segments are displayed as a table.

Unlike descriptive flexfields, the extension columns corresponding to extensible flexfields segments are part of extension tables, separate from the base application table. Unlike descriptive flexfield contexts, the set of attributes in an extensible flexfield context remains constant and does not differ by context value.

An extensible flexfield describes an application entity, with the runtime ability to expand the database that implementation consultants can use to define the data structure that appears in the application.

Extensible flexfields support one-to-many relationships between the entity and the extended attribute rows.

For lists of extensible flexfields, see assets with the Flexfield: Extensible type in Oracle Enterprise Repository for Oracle Fusion Applications (<http://fusionappsoer.oracle.com>)

Usages

As with descriptive flexfields, you can define multiple usages for an extensible flexfield, which enables several application tables to share the same flexfield.

For example, a flexfield for shipping options can be used by both a Supplier table and a Buyer table. In addition, you can associate a context with one, some, or all of the flexfield's usages. Thus, with the shipping information example, you can associate a warehouse context with the Supplier usage, a delivery location context with the Buyer usage, and a ship-via context with all usages.

Categories

Extensible flexfields support multiple contexts, and the contexts can be grouped into categories. Every extensible flexfield has at least one category, or root category, and some extensible flexfields support a hierarchy of categories, where a given category can inherit contexts from its parent categories.

You can define categories for extensible flexfields, and you can associate any combination of contexts with a given category.

For example, the Electronics and Computers category hierarchy might include a Home Entertainment category, which in turn might include an Audio category and a TV category, and so on. The Home Entertainment product might have contexts that specify voltage, dimensions, inputs and outputs. Contexts are reusable within a given extensible flexfield. For example, the dimensions context could be assigned to any category that needs to include dimensional information.

Pages

Extensible flexfields allow you to combine contexts into groups known as pages, which serve to connect the contexts so they will always be presented together in the application user interface.

Each application page corresponds to one extensible flexfield category, with a separate region of the page for each associated context.

Managing Extensible Flexfields: Points to Consider

Configuring extensible flexfields involves managing the available flexfields registered with your application database and configuring their flexfield-level properties, defining contexts, categories, and pages, and configuring the segments for each extensible flexfield.

Contexts

A context can be defined as single row or multi row. Single row contexts are the same as descriptive flexfields contexts. A single row context has only one set of context-sensitive segments. A multi-row context enables you to associate multiple sets of values with the same object instance.

For example, for a BOOK table, you could create a multi-row context named chapters that contains a chapter segment and a number of pages segment. Multiple chapters can then be associated with each book in the BOOK table.

Set the context to translatable so free-form text entered by end users is stored in the language of the user's locale, and different translations of that text can be stored in other languages. Segments in the translated contexts should utilize format-only value sets for storing free-form, user-entered text.

Categories

A category is a grouping of related data items that can be considered to belong together. You can associate any combination of contexts with a given category. Extensible flexfields with more than 30 categories must be deployed using the Deploy Offline command.

A category hierarchy logically organizes a set of categories.

For example, the Electronics and Computers category hierarchy might include a Computer category and a Home Entertainment category, which in turn might include an Audio category and a TV category, and so on.

A category can be a child or sibling of an existing category. The hierarchy can be as simple or as complex as desired, with any combination of zero or more sibling categories and zero or more child categories. If no category is defined, the data items are grouped under a single predefined default category.

Each category has associated contexts that store relevant information about a data item in that category. For example, a Home Entertainment product has contexts that specify Voltage, Dimensions, Inputs and Outputs. Contexts are reusable within a given extensible flexfield; the Dimensions context could be assigned to any category that needs to include dimensional information.

If a hierarchy includes child categories, each child category inherits the contexts from its parent category; for example, the Home Entertainment category inherits Voltage and Dimensions from the Electronics and Computers category.

Each extensible flexfield is associated with a particular category hierarchy. You can think of category hierarchies as the defining framework for extensible flexfields and their contexts. A category hierarchy specifies which contexts are valid for each category.

An extensible flexfield can include multiple contexts which you define to support a given category. These contexts can be suitable for a variety of different purposes, but within a particular category, some contexts might be considered to be related to, or dependent on, each other. You can combine these contexts into groups known as pages, which serve to connect the contexts so they will always be presented together in the application user interface.

For example, the Home Entertainment category might have an Electrical Specifications page that contains the Voltage, Inputs and Outputs contexts, and a Physical Specifications page that contains the Dimensions and Form Factor contexts.

Indexed Segments

You can designate an extensible flexfield segment as indexed so that it is one of the selectively required attributes an end user can use in an attribute search. If

you indicate in the Manage Extensible Flexfield UI page that a segment should be indexed, the column representing the segment must be added to the database index. This is commonly done by a database administrator (DBA).

When an extensible flexfield with indexed segments is deployed, search task flows are generated along with the other flexfield artifacts and specify the indexed attributes as selectively required. In the deployed extensible flexfield's search task flow, an end user must specify at least one of the indexed attributes in the search criteria. This prevents non-selective searches which could cause performance issues.

For example, if you index the memory and processor attributes and ensure that the corresponding columns in the database are indexed, an end user can search an item catalog for computers by entering processor or memory or both as a search criterion. No search is performed if an end user enters an attribute as search criteria that is not indexed.

Offline Deployment

You can deploy extensible flexfields offline as a background process using the Deploy Offline command in the Manage Extensible Flexfields task, and continue working in the session without having to wait for the deployment to complete. Deploy Offline enables adding one after another extensible flexfield to your deployment queue. For extensible flexfields with more than 30 categories you must use Deploy Offline.

You can remove an extensible flexfield from the deployment queue with the Cancel Offline command.

When an extensible flexfield is deployed offline, its offline status indicates that the flexfield is in an offline deployment process. The Offline Status column refreshes when you perform a new search in the Manage Extensible Flexfields task. A flexfield's offline status is cleared and its deployment status updated when the offline deployment process has completed.

Manage Key Flexfields

Key Flexfields: Explained

Key flexfields provide a means to capture a key such as a part number, a job code, or an account code. A key flexfield consists of one or more segments, where each segment can have a meaning.

For example, a part number 10-PEN-BLA-450 might correspond to a black pen from vendor #450 sold by division #10 (office supplies). Behind the scenes, the application uses a unique number, 13452, for this part, but the end user always see the 10-PEN-BLA-450 part number.

The following aspects are important to understanding key flexfields.

- Architecture
- Segments and segment labels
- Structures
- Segment and structure instances

- Combinations
- Dynamic combination creation
- Security

Key flexfields are not optional. You must configure key flexfields to ensure that your applications operate correctly. You configure and maintain key flexfield definitions with the Manage Key Flexfields task.

For lists of key flexfields, see assets with the Flexfield: Key type in Oracle Enterprise Repository for Oracle Fusion Applications (<http://fusionappsoer.oracle.com>).

Architecture

When you configure a key flexfield, you define metadata about the key flexfield such as how many segments are in a structure, how many structures the flexfield uses, what value sets each segment uses, and so on. This is flexfield metadata stored in flexfield metadata tables.

Based on the flexfield metadata, actual part numbers are captured at runtime as a combination of segment values and stored in a combinations table. A combinations table contains all the segment columns for a flexfield, plus a unique ID column and a structure instance number column that differentiates multiple arrangements of the segment columns.

For example, a part number that can be comprised of multiple segments can be represented by a key flexfield. A part number key flexfield has a corresponding combinations table, where the flexfield stores a list of the complete codes, with one column for each segment of the code, together with the corresponding unique ID and structure instance number for the code. When users define a new part number or maintain existing part numbers in the parts catalog, they directly maintain rows in the combination table.

The foreign key table contains a different business entity than the combinations table. For example, the business entity in the foreign key table is order lines or invoice lines that contain foreign key references to parts for ordering and so on. Any number of foreign key tables can reference a particular entity represented by a key flexfield.

Segments and Segment Labels

A key flexfield consists of segments. Segments consist of a prompt, a short prompt, display width, a number that determines where in the sequence of a key flexfield structure the segment exists, the range type and the column name of the attribute being captured by the segment, a default value set and a label for the segment. A segment label identifies a particular segment of a key flexfield. Segment labels are defined and made available by applications development.

Applications identify a particular segment for some purpose such as security or computations. Segment name or segment order cannot reliably identify a segment because key flexfield segments can be configured to appear in any order with any prompts. A segment label functions as a tag for a segment.

For example, Oracle Fusion General Ledger needs to identify which segment in the Accounting Flexfield contains balancing information and which segment contains natural account information. General Ledger uses a segment label to

determine which segment you are using for natural account information. When you define your Accounting Flexfield, you must specify which segment label apply to which segments.

Some labels must be unique, and cannot be applied to more than one segment in each structure. Other labels are required, and must be applied to at least one segment in each structure.

A segment label orients an end user's search of segments, such as the Cost Center label for all segments across key flexfields that capture a value for cost center.

Structures

A key flexfield structure definition includes the number of segments and their order.

In some applications, different users need to see different segment structures for the same flexfield. A key flexfield can have multiple structures if registered to support more than one structure.

The flexfield can display different fields for different end users based on a data condition in your application data, such as the value of another field entered by the end user or the user's role. For example, the correctly formatted local postal address for customer service inquiries differs based on locale. A postal address key flexfield could display different segments and prompts for different end users based on a location condition in your application data, such as the user's role or a value entered by the user.

Each structure can have one or more segments. Thus a segment is a child of a structure. If you want to store a particular segment, such as Cost Center, in two different structures, you must define the segment separately in each structures.

Each structure may have one or more structure instances. Each instance of a structure shares the same number and order of segments, but differs in the allowable values or value sets that validate the segments.

Structure and Segment Instances

You can define multiple configurations of a key flexfield structure. These structure instances have the same segment structure, in the same sequence order. They differ primarily in how each segment is validated. You define a structure instance for each key flexfield and each key flexfield structure instance.

The segments in a key flexfield structure instance are segment instances. A segment instance is a segment with a specific value set assigned to it.

If a key flexfield has been registered with a tree structure, you can specify a tree code for a segment instance, where the tree code defines a hierarchical relationship between the segment values.

Combinations

A combination is a complete code, or combination of segment values that makes up the code, that uniquely identifies an object.

For example, each part number is a single combination, such as PAD-YEL-11x14 or 01-COM-876-7BG-LTN. In these combinations, the hyphen is the segment separator. If you had ten parts you would define ten combinations. A valid

combination is simply an existing or new combination that can currently be used because it is not out of date or disabled, and does not violate cross-validation or security rules. A combination has different segments depending on the flexfield structure being used for that combination. Any combination is associated with only one particular flexfield structure.

Many Oracle Fusion Applications products refer to a key flexfield combination by using the name of the entity or the key flexfield itself. For example, Oracle Fusion Assets uses the asset key flexfield and refers to one of its combinations as an asset key or asset key flexfield. In another example, other Oracle Fusion Applications products including Oracle Fusion General Ledger (GL) refer to combinations of the accounting flexfield as account or GL account.

Each key flexfield has one corresponding table, known as the combinations table, where the flexfield stores a list of the complete codes, with one column for each segment of the code, together with the corresponding unique ID number (a code combination ID number or CCID) for that code. Then, other tables in the application have a column that stores just the unique ID for the code. For example, you may have a part number code, such as PAD-YEL-11x14. The Parts combinations table stores that code along with its ID, 57494. If your application allows you to take orders for parts, you might then have an Orders table that stores orders for parts. That Orders table would contain a single column that contains the part ID, 57494, instead of several columns for the complete code PAD-YEL-11x14.

Typically one combinations page maintains the key flexfield, where the key flexfield is the representation of an entity in your application. The combinations page is where you maintain individual combinations, such as part numbers.

Dynamic Combination Creation

Dynamic combination creation is the insertion of a new valid combination into a combinations table from a page other than the combinations page.

Dynamic combination creation may be enabled at the following levels.

Level Of Dynamic Combination Creation	Controlled By:
Flexfield	Application development
Each usage or reference to the key flexfield	Application development
Structure instance	Administrators and implementation consultants
Other	Administrators and implementation consultants

If your key flexfield or certain usages or references of the key flexfield do not permit dynamic combination creation, you may control whether dynamic combination creation is enabled for each structure instance. If enabled, a user can enter a new combination of segment values using the flexfield window from a foreign key page. For example, when entering a transaction, a GL user can enter a new expense account code combination for an account that does not yet exist. Your application creates the new account by inserting the new combination into the combinations table behind the scenes. Assuming that the new combination satisfies any existing cross-validation rules, the flexfield inserts the new combination into the combinations table, even though the combinations table is not the underlying table for the foreign key page.

Managing Key Flexfields: Points to Consider

Consider the plans for a key flexfield, security, and resulting runtime pages when configuring key flexfields.

Planning

Plan structures carefully and allow for future needs.

Caution

Do not change the number, order, and maximum length of segments once you have acquired flexfield data.

Structure Delimiters

A delimiter separates the segments when they appear to end users. The delimiter value of a structure specifies the character used to visually separate segment values when the key flexfield is displayed as a string of concatenated segments in the UI.

Tip

Choose the delimiter value of your key flexfield carefully so that it does not conflict with the flexfield data. For example, if your data frequently contains periods, such as in monetary or numeric values, do not use a period as your segment separator. Any character you expect to appear frequently in your segment values or descriptions is not a good choice for the delimiter.

If you change the configuration of a key flexfield, such as the delimiter, the change affects the previously stored key flexfields with that structure.

Security

Oracle Fusion data security enforces value set security.

Within key flexfields, value set security applies to the selection of the individual segment values in the segment list of values. When selecting a key flexfield segment value from the combination table, data security allows display of only the combinations whose segment values you have access to. Applications development controls whether or not value set security rules propagate to the foreign key table. By default they do.

Runtime Pages

Application development determines the user interface (UI) pages used to render flexfields. The types of key flexfield UI pages are as follows.

- Combinations pages where underlying entity objects use the combinations table itself
- Foreign key pages where the underlying entity objects contain a foreign key reference to the combinations table
- Partial usage page where some or all of the key flexfield's segment columns are in a product table

The same key flexfield can be used in different ways on different pages.

A page with a foreign key reference has a base table or view that contains a foreign key reference to a combinations table with the actual flexfield segment columns. This allows manipulating rows containing code combination IDs (CCID).

A page with partial usage of a key flexfield presents segments that are defined on a product's transactional table in addition to being defined on a combinations table. In the case of a partial usage page, it is possible that only part of the configuration is visible. This allows the key flexfield to behave more like a descriptive flexfield.

A code combination maintenance page or combinations page presents the combinations table. This allows directly creating and maintaining code combinations. The combinations table contains all key flexfield segment columns and a unique ID column.

A typical application has one and only one combinations page. An application might not have a combinations page if it does not support maintenance by administrators.

A page containing a search region enables end users to select which attributes of the key flexfield view object to use as criteria to search for flexfield metadata.

For example, you can configure seven segments for the Account key flexfield. In a foreign key reference page, end users see the typical key flexfield picker with all seven segments where they can search for combinations. In a partial usage page using the same key flexfield, end users potentially could see only a single segment such as the Cost Center labeled segment, or they might see multiple segments but displayed as individual segments rather than as a picker for choosing combinations.

For more information on key flexfield pages, see the Oracle Fusion Applications Developer's Guide.

Key Flexfield Structures: Explained

A key flexfield structure arranges the segments of a key so you can reuse a single key flexfield in multiple combinations of the same or a subset of segments. Multiple instances of a single structure can accommodate differences in the value sets assigned to the structure's segments.

The structure determines the following aspects of a key flexfield.

- The segments to include
- The order of the segments
- Segment labels on the included segments
- Properties for each segment applied to the instances of the segments in an instance of the structure

Managing Key Flexfield Structures

All the segments defined for a key flexfield are available to be included in a key flexfield structure.

You can define as many segments as there are defined segment columns in your key flexfield combinations table.

Restriction

Be sure to add segments in the order that your key requires. Once deployed, the order cannot be changed.

Enable segments to indicate that they are in use. A flexfield does not display disabled segments in runtime.

Tip

To protect the integrity of your data, disable a segment if you have already used it to enter data.

Key Flexfield Structure Instances and Segment Instances: Explained

A key flexfield structure can have one or more alternate structure instances.

The instances of a key flexfield structure share the following aspects of the structure.

- The same set of segments
- The same arrangement of segments
- The same properties at the segment and structure levels

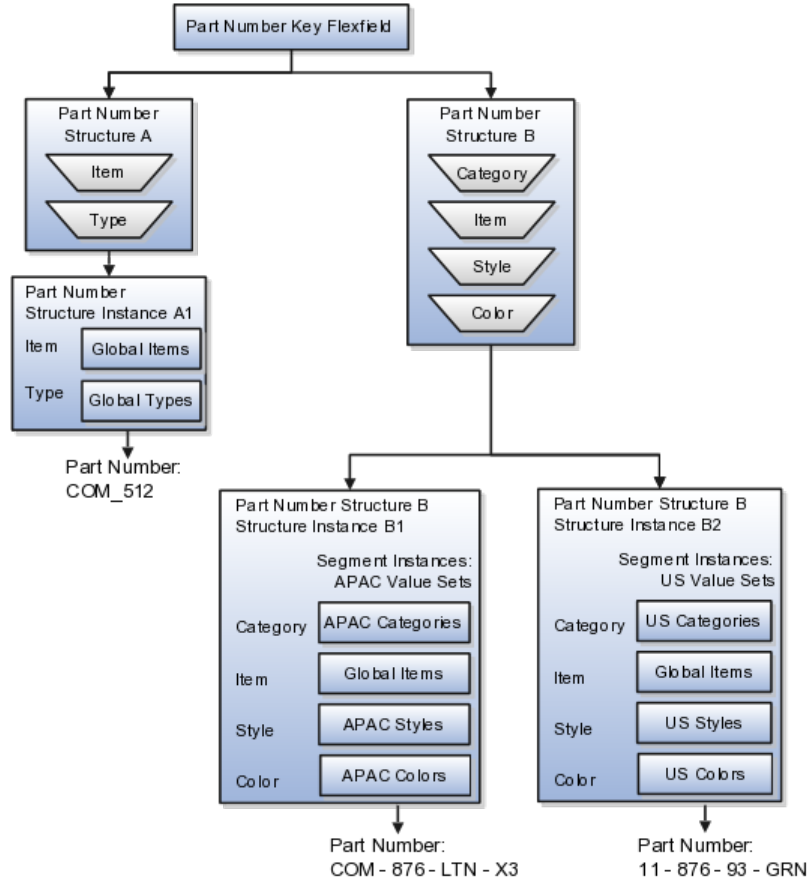
Differences among structure instances at the structure level include whether dynamic combination creation is allowed.

Differences among segment instances at the structure instance level include the following.

- Value set
- Default type and default value
- Tree code
- Whether the segment is any of the following
 - Required
 - Displayed
 - Enabled for business intelligence
 - Optional or required as a query criterion

For example, you could use one group of value sets for the US and another for France.

The figure shows two structures instances for a part number structure. The structures differ in the number of segments and the segment separators used. The structure instances of a structure share all properties that are defined for the structure, but can vary in the properties defined at the structure instance or segment instance level, such as the value set assigned to the segment instances.



Query Required Segment Instances

You can designate a key flexfield segment instance as query required so that it is one of the selectively required attributes an end user can use in a key flexfield combination search. If you indicate in the Manage Key Flexfields UI page that a segment instance should be indexed, the column representing the segment must be added to the database index. This is commonly done by a database administrator (DBA).

Following deployment, the combination picker of the key flexfield displays the query required attributes as selectively required. An end user must specify at least one of the query required attributes in the search criteria. This prevents non-selective searches that could cause performance issues.

For example, if you mark the cost center and account attributes as query required and ensure that the corresponding columns in the database are indexed, an end user can search for combinations by entering cost center or account or both as a search criterion. No search is performed if an end user does not enter at least one query required attribute as search criteria.

Tip

Index the Structure Instance Number column on your combinations table to improve performance.

Dynamic Combinations

If a key flexfield supports dynamic combination creation, you can choose to enable this feature by selecting **Dynamic Combination Creation Allowed**. This will allow end users to enter values at runtime that produce new code combinations for the flexfield. If not enabled, new valid combinations can only be entered using the combinations table for the flexfield.

Trees

If a tree code has been defined for the value set assigned to the segment instance, and you assign the tree code to the segment instance, tree hierarchy search operations are available on the segment values.

For a segment instance to be based on a tree, the following must be true.

- Application development registered the key flexfield with a tree structure.
- A tree code for that tree structure exists.
- The tree code that includes tree versions containing the values of the value set assigned to the segment instance.
- You assign the desired tree code directly to the segment instance.

Provided these conditions are satisfied, different segment instances that use the same value set can be assigned the same or different tree codes, meaning they use a different hierarchy definition over the same values.

Enabling Key Flexfield Segments for Business Intelligence: Points to Consider

A key flexfield that is registered in the database as enabled for Oracle Business Intelligence (BI) includes a BI Enabled setting for each of its segment instances. When a segment instance is BI-enabled, it is available for use in Oracle Business Intelligence.

The following aspects are important in understanding BI-enabled key flexfield segments.

- Flattening business components to use BI-enabled segments in Oracle BI.
- Equalizing segments to prevent duplication and complexity in the flattened component.
- Mapping attributes of flattened business components to logical objects in Oracle BI.
- Managing the labels that map segments to logical objects in Oracle BI.

After you deploy a business intelligence-enabled flexfield, import the flexfield changes into the Oracle Business Intelligence repository to make use of the newly generated attributes in business intelligence applications. For additional information about import, refer to the Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition).

Flattening

When you deploy a business intelligence-enabled key flexfield, the deployment process generates an additional set of flattened business components for use in business intelligence. The flattened business components include attributes for business intelligence-enabled segment instances only.

If you assigned a label to a segment, the flattened components include a single attribute representing all segment instances with that label. If you did not assign a label, the flattened components include a discrete attribute for each BI-enabled segment instance in each structure.

Equalization

You can prevent duplication and the extra workload and complexity that result from the flattening process by assigning a label to any set of segments that serve the same purpose in different structures. BI-enabled segments that have labels are equalized so that the flattened business components include one attribute for each label, consolidated across structures. Non-labeled segments are not equalized across structures, so the flattened components include a separate attribute for each segment for each structure.

Note

It may not be possible to equalize similarly labeled segments if they have incompatible data types or value set types.

Mapping to Logical Objects in Business Intelligence

Assign a label to a segment to map the corresponding attribute in the flattened components to a logical object in Oracle Business Intelligence. Using labels to map segments to BI logical objects minimizes the steps for importing the flexfield into Oracle Business Intelligence.

Note

Assigning a label to a segment serves to equalize the attribute across structures, as well as map the equalized attribute to business intelligence.

Managing Labels

You may assign a predefined label (if available) to segments or create new labels for assignment, as needed. Specify a code, name, and description to identify each label. In the BI Object Name field, enter the name of the logical object in Oracle Business Intelligence to which the segment label should map during import. Specifying the BI logical object minimizes the steps for importing the flexfield into Oracle Business Intelligence and helps to equalize context-sensitive segments across structures.

If no labels are assigned to a BI-enabled segment, or the BI Object Name on the assigned label does not exist in business intelligence, you must manually map the segment to the desired logical object when importing into Oracle Business Intelligence.

In addition, segments without labels cannot be equalized across structures. The flattened components include a separate attribute for each non-labeled segment in each structure.

Note

Segment labels serve other functions as well, as presented in Key Flexfields: Explained.

Key Flexfields: Example

A key flexfield can capture expense account information.

Scenario

When entering details for each expense, the user specifies an account to which the expense is charged.

Entering Expense Accounts

A user interface for entering expenses gives the user the option of selecting an expense account that identifies the cost center and other details needed for processing the expense.

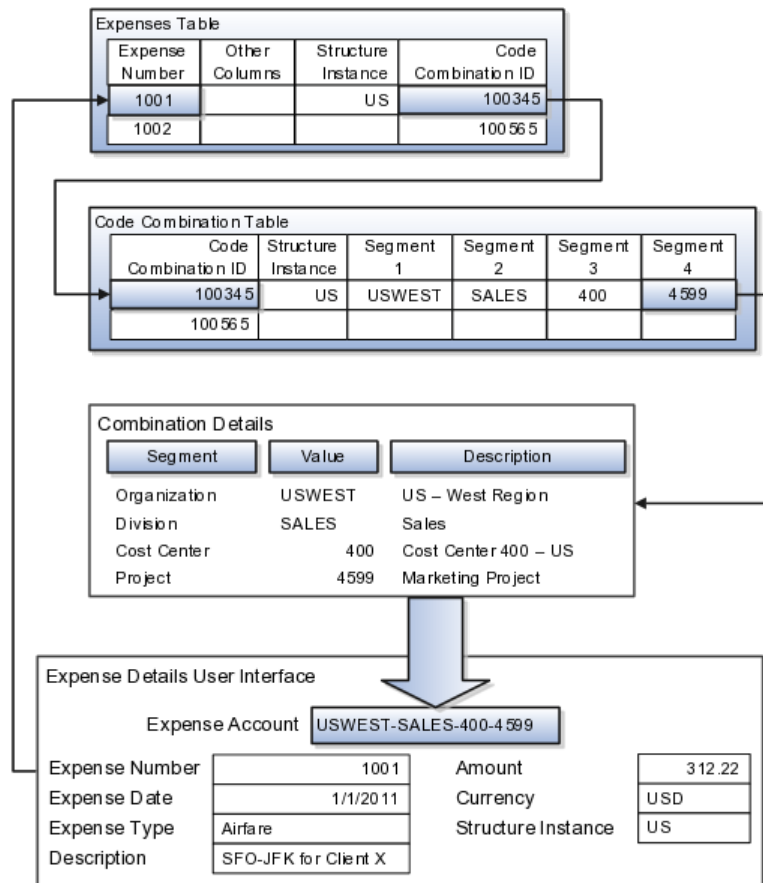
Analysis

The expense account field is a foreign key reference to a code combination (EXPENSE_LINES.EXPENSE_ACCOUNT = ACCOUNTS.CCID).

Code Combination Table for Entering Accounts and Employees

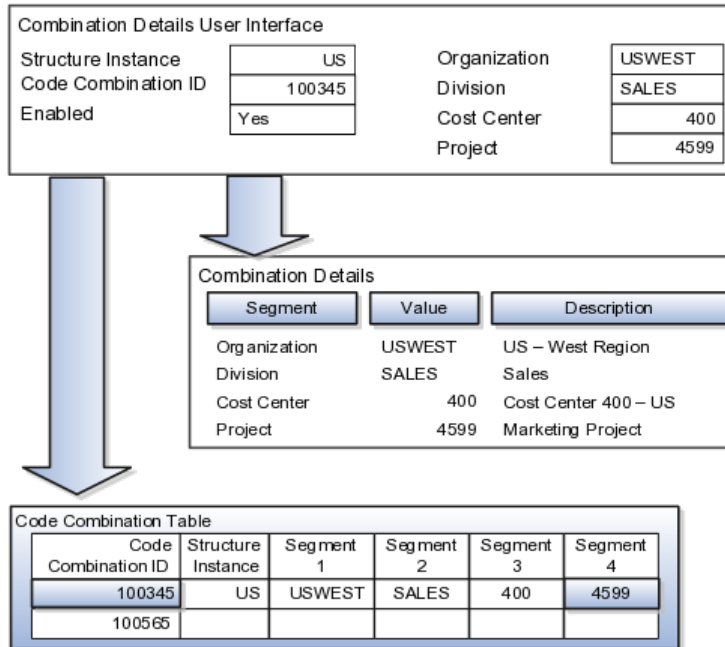
The code combination table supports entering account information, such as for expense accounts.

The figure shows the origin in the code combination table of the account specified by the user. The code combination ID record stores the information of the key flexfield segments used to assemble the expense account based on the key flexfield configuration.



The combinations page, which is the maintenance page for the key flexfield, is for managing rows in the combination table. In this example, managing the combinations means adding or editing account numbers that adhere to the key flexfield metadata rules.

The figure shows the code combination details for the example expense account reflected in the flexfield configuration and the code combination table.

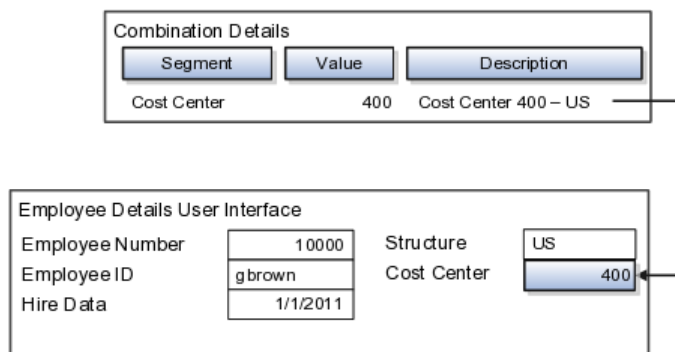


If dynamic combination creation is not enabled, then when entering an expense line, the user can only select an account that already exists in the ACCOUNTS (combinations) table. If they require an account that does not exist, they must consult with the appropriate application administrator who can add the account to the combinations table.

If dynamic combination creation is enabled, then when entering an expense line, the user can either select a pre-existing account, or type in a new account that created dynamically on the fly in the ACCOUNTS (combinations) table. Once the new combination is created, the same user can refer to it on the expense line.

When managing employee information, the user specifies the cost center that the employee belongs to. The cost center field corresponds to a single, labeled segment of the Account Key Flexfield and has metadata defined such as the allowable value set for that segment.

In this figure, instead of specifying a cost center ID reference to an account, only the Cost Center segment is used and the value is stored directly on the employee table.



FAQs for Define Flexfields

Why did my flexfield changes not appear in the runtime UI?

The ADF business components or artifacts of a flexfield, which are generated into an metadata services (MDS) repository when the flexfield is deployed, are cached within a user session. You must sign out and back in again to view flexfield definition changes reflected in the runtime application user interface page.

A flexfield's status relative to its deployment determines whether the flexfield segments as currently defined in the metadata are available to end users. The flexfield segments seen by end users in the runtime correspond to the flexfield definition that was last deployed successfully.

What happens if a value set is security enabled?

Value set security is a feature that enables you to secure access to value set values based on the end user's role in the system.

As an example, suppose you have a value set of US state names. When this value set is used to validate a flexfield segment, and users can select a value for the segment, you can use value set security to restrict them to selecting only a certain state or subset of states based on their assigned roles in the system.

For example, Western-region employees may choose only California, Nevada, Oregon, and so on as valid values. They cannot select non-Western-region states. Eastern-region employees may choose only New York, New Jersey, Virginia, and so on as valid values, but cannot select non-Eastern-region states. Value set security is implemented using Oracle Fusion Applications data security.

How can I set a default value for a flexfield segment?

When you define or edit a flexfield segment, you specify a default value from the values provided by the value set assigned to that segment.

You can set the default value to be a parameter, which means the entity object attribute to which the parameter you choose is mapped will provide the initial default value for the segment.

You can set to be a constant, if appropriate to the data type of the value set assigned to the segment.

In addition to an initial default value, you can set a derivation value for updating the attribute's value every time the parameter value changes. The parameter you choose identifies the entity object source attribute. Any changes in the value of the source attribute during runtime are reflected in the value of the segment.

If the display type of the segment is a check box, you can set whether the default value of the segment is checked or unchecked.

Define Attachments

Attachments: Explained

Attachments are pieces of supplementary information that users can associate with specific business objects such as expense reports or purchase orders. Attachments can be URLs, desktop files, text, or in cases where available, repository folders. For any given business object, a user may be able to only view attachments, or also create, delete, or edit attachments, depending on security. For more information on an introduction to attachments, see the Oracle Fusion Applications Developer's Guide.

Repository

Attachments are stored in a content 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.

Action	Privilege
View	Read Application Attachment (FND_READ_APPLICATION_ATTACHMENT_DATA)
Add or Update	Update Application Attachment (FND_UPDATE_APPLICATION_ATTACHMENT_DATA)
Delete	Delete Application Attachment (FND_DELETE_APPLICATION_ATTACHMENT_DATA)

For example, if users have the Read Application Attachment privilege for all categories associated with the expense report attachment entity, except the Receipts attachment category, then they can view all expense report attachments

except those created with the Receipts category. Likewise, if users do not have the Update Application Attachment privilege for any attachment categories tied to the expense report attachment entity, then they cannot create any attachments at all for expense reports.

For more information on attachment category data security, see the Oracle Fusion Applications Developer's Guide.

Finally, certain attachments UI for users have predefined restrictions on categories in place. Your developers can also introduce additional filters to determine which document categories are available for a specific page. Check with your developers or help desk.

Issue: Missing Attachment Category

Users can see existing attachments, but the attachments no longer have an attachment category value.

Resolution

When the attachment was added, at least one category existed for the corresponding attachment entity, as otherwise the attachment could not have been added. Since then, the entity was edited so that it no longer has any assigned categories, so the user cannot see the category associated with that attachment.

Use the Manage Attachment Entities page to reassign attachment categories to the relevant attachment entity. For example, if users can no longer see the Receipts attachment category for an attachment to an expense report, then search for the expense report attachment entity and assign to it the Receipts category. You might need to check with your system administrator or help desk to determine the exact entity used on the page with the expenses attachments or what additional categories to assign.

Finally, certain attachments UI for users have predefined restrictions on categories in place. Your developers can also introduce additional filters to determine which document categories are available for a specific page. Check with your developers or help desk.

FAQs for Define Attachments

What's an attachment category?

An attachment category is used to classify and secure attachments. Each attachment user interface must be defined with at least one category for users to be able to add attachments. If there are multiple categories, users can view them and select one when adding attachments. For example, attachments for an expense report can be categorized as receipts, scanned invoice images, and so on.

You can also associate roles with categories to determine user access and actions for attachments, based on the categories assigned to the attachment entity. For example, security for expense report attachments can be based in part on the categories assigned to the expense report attachment entity. You can define multiple categories per module, and add and manage custom categories for your own purposes. For more information on attachment category data security, see the Oracle Fusion Applications Developer's Guide.

Use the Manage Attachment Categories page, which you can access by starting in the Setup and Maintenance Overview page and searching for the Manage Attachment Categories task.

Set Activity Stream Options

Activity Stream Options: Highlights

Activity Stream is a region on the Oracle Fusion Applications Welcome dashboard and other pages in various applications. Users track the activities and transactions of other users in this region. You can set options that affect the all Activity Stream regions for all users across your site. Individual users can still override your settings through Activity Stream preferences.

Activity stream settings are described in the Oracle Fusion Middleware User's Guide for Oracle WebCenter Portal: Spaces. When you read content from that guide, note that:

- Your setup applies to all users, not just yourself or any individual user.
- You can disregard discussions about how to access the settings, because you access the Set Activity Stream Options page by starting in the Setup and Maintenance Overview page and searching for the Set Activity Stream Options task.

Setting Activity Stream Options

- Define the types of users to display activities about in the Activity Stream region, the types of activities to track, and other settings.

See: Setting Activity Stream Preferences

Manage Menu Customizations

Managing Menu Customizations: Highlights

You can customize the Navigator menu, which is the main menu of Oracle Fusion Applications and is always available in the global area. You can also customize the home page of Oracle Fusion Applications by adding tabs to display additional dashboards. Use the Manage Menu Customizations page, which you can access by starting in the Setup and Maintenance Overview page and searching for the Manage Menu Customization task.

An overview of customizing the Navigator menu and home page is provided in the Oracle Fusion Applications Extensibility Guide.

Navigator Menu and Home Page

- Navigator menu customization involves managing items, which are nodes in the menu that take the user to the desired destination, and groups, which are categories of items.

See: Customizing the Navigator Menu

- The Oracle Fusion Applications home page displays a set of dashboards as tabs. You can create more tabs and make them display additional dashboard content.

See: Customizing the Navigator Menu.

Manage Audit Policies

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.

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.

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://fusionappsoer.oracle.com>.

Oracle Fusion Middleware Products

- Configure business objects to enable auditing in Oracle Fusion Middleware products. Refer to the Oracle Fusion Middleware Security and Administrator's Guide for Web Services.

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 Oracle Social Network to collaborate with key individuals within your organization, as well as external contacts such as suppliers and partners, using critical transaction data from Oracle Fusion Applications. Available collaboration tools include groups, activity feeds, discussion forums, document sharing, instant messaging, email and web conferencing. By bringing key attributes from the applications to share, socialize and update information, you can make better business decisions, based on additional information that you obtain and analyze within your social network environment. You do this by defining which business objects and attributes to enable and the enablement method for Oracle Social Network integration with Oracle Fusion Applications. Once enabled, these business objects can be shared in collaborative conversations. Within Oracle Fusion Applications, conversations can be viewed using the Social button in the Global Area and select applications pages.

Use the Manage Oracle Social Network Objects page, which you can access by starting in the Setup and Maintenance Overview page and searching for the task Manage Oracle Social Network Objects.

Note

Oracle Social Network is currently not available in non-Cloud implementations.

Aspects of managing Oracle Social Network objects include the following:

- Enabling business objects for integration.
- Understanding initial and subsequent defaults.

Enabling Business Objects for Integration

A business object cannot be shared within Oracle Social Network until a functional administrator or implementor accesses the Manage Oracle Social

Network Objects page and enables the business object for Oracle Social Network integration. Use the Manage Oracle Social Network Objects page to set up and define which business objects and attributes to enable and the enablement method for Oracle Social Network integration with Oracle Fusion Applications.

Note

You see only the business objects that are predefined as available to enable, not the full list of business objects for products that you are permitted to view.

Understanding Initial and Subsequent Defaults

Upon initial deployment, the preceding rule that states a business object cannot be shared until it is accessed and enabled through the Manage Oracle Social Network Objects page, does not apply to certain default configurations for business objects in Oracle Fusion Customer Relationship Management (CRM) and Oracle Fusion Human Capital Management (HCM). Subsequently, these business objects are managed no differently than any of the other business objects. If you set to Manual or Automatic, the setup is not complete until you select attributes.

Options for Enabling Oracle Social Network Objects: Explained

Access the Manage Oracle Social Network Objects page to select from the following options to enable business objects and apply attributes for Oracle Social Network integration with Oracle Fusion Applications.

These enablement options exist:

- Enable Object
- Disable Object
- Enable All

Enable Object

Access the Business Object section of the Manage Oracle Social Network Objects page.

Click **Enable Object** to select from the available enablement options and define how the object integrates with Oracle Social Network. Options include:

- **Manual** - Recommended; the object is enabled for manual sharing with Oracle Social Network, where the social network user decides whether or not to share each instance of the object with Oracle Social Network. Once shared, all updates to enabled attributes of the object instance, and deletes, are sent to Oracle Social Network. Updates to attributes that are not enabled are not sent.
- **Automatic** - Automatically sends newly enabled object instances and updates to Oracle Social Network. All object instances are automatically shared with Oracle Social Network upon creation and all subsequent updates to enabled attributes of the object instances, and deletes, are automatically sent to Oracle Social Network.

- **No** - Does not send any object instance information to Oracle 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 Oracle Social Network.

Disable Object

Access the Business Object section of the Manage Oracle Social Network Objects page.

Click **Disable Object** to disable the selected business object, then Save. This shortcut disables the currently selected business object by changing the enablement option to No.

Alternatively, you can click **Enable Object** to access the dialog box and enablement options and select **No**, then click OK, and then click Save to disable a selected business object.

Disabling the object through either mechanism does not affect any associated attributes. Use the Attributes section of the page to manage the list of enabled attributes.

Enable All

Access the Manage Oracle Social Network Objects page.

Click **Enable All** at the page level to perform a bulk update of the enablement status from No to Manual of all business objects. After you click Enable All, then you save.

The bulk update of the statuses does not affect the attributes selected as enabled for each business object. Use the Attributes section of the page to manage the list of enabled attributes.

Status Column

The Status column appears in the Business Objects table to provide a visual indicator as to which enabled business objects do not yet have an enabled attribute assigned.

When you enable a business object (automatic or manual), a check mark or a warning sign appears in the Status column of the business object depending on this criteria:

- A check mark indicates that you have configured attributes for an enabled business object.
- A warning sign indicates that you have not configured any attributes for the enabled business object.

Note

It is important that you configure the attributes before you click Save.

When you enable a business object, but do not configure any attributes for the enabled business object, unexpected results can occur as no attributes are sent to Oracle Social Network during create and update except some internal bookkeeping information. Deletes are sent as usual.

Attributes

Enable business object attributes in the Attributes section of the Manage Oracle Social Objects page. To provide a visual indicator of the enabled business objects that have no attributes added, a Status column appears at the end of the Business Objects table. This column shows a warning when there are no attributes defined for a business object that has had its enabled state changed from No. This warning does not prevent you from saving the Oracle Social Network enablement of the business object, but does provide a guide that you should assign attributes for an enabled business object.

In the Attributes table, you can do the following:

- View a list of all attributes that are enabled.

These are the attributes that, upon update, are sent to Oracle Social Network. Disabled attributes are not sent.
- Click Add to launch the dialog where you can select attributes to add to the table.
- Click Remove to remove attributes from the table.

FAQs for Manage Oracle Social Network Objects

What happens if I update translations?

When you click **Update Translations** from the Manage Oracle Social Network Objects page, you synchronize the newly translated text from Oracle Fusion Applications so that it can be used within Oracle Social Network. This means you can install and enable a new language or take a language patch at any time.

Update Translations sends attribute labels and business object names to Oracle Social Network for use in its user interface. Users in Oracle Social Network see the attribute or business object labels in the language of their locale. If the user changes locale in Oracle Social Network, then the attribute or business object labels appear in the updated language. The data, however, appears in the language in which it was originally sent to Oracle Social Network. If a user has previously sent a business object instance to Oracle Social Network, then the business object instance data is not updated. When you save the business object enablement to Oracle Social Network, it sends the translations as well. This makes it unnecessary to click both Save and Update Translations. For subsequent updates to labels and attributes, such as if a new Oracle Fusion Applications language pack is installed, run Update Translations to send the updated labels and their translations to Oracle Social Network. The Update Translations button sends translations for all business objects whose enablement is Manual or Automatic.

Manage Applications Core Common Reference Objects: Manage Applications Core Messages

Common Messages: Points to Consider

Common messages, which have message names that begin with FND_CMN and message numbers between 0 and 999, are used throughout Oracle Fusion Applications. Each common message can appear in multiple places in any product family. For example, the FND_CMN_NEW_SRCH message can be used for any search to indicate that no results were found. Common messages that are of type error or warning are part of the message dictionary.

Editing Common Messages

Because a common message can be used in any application, consider the ramifications if you edit any aspect of the message, including incident and logging settings. Changes would be reflected in all instances where the message is used. For example, if you change the message text, make sure that the text would make sense to all users across Oracle Fusion Applications who might see it.

Creating Common Messages

You can create custom common messages for use in multiple places within a single product. Do not begin the message name with FND_CMN, but use another suitable convention. The message number should be within the range that is designated for the product.

Manage Applications Core Common Reference Objects: Manage Applications Core Administrator Profile Values

Creating and Editing Messages: Highlights

Each message in the message dictionary has many attributes and components, including message properties, text, and tokens, that you define when creating or editing the message.

Details about these messages are described in the Oracle Fusion Applications Developer's Guide.

Message Properties

- The message type identifies the type of information that the message contains.

See: Understanding Message Types

- The message name and number are identifiers for the message. There are specific message number ranges for predefined messages in each application, and you should not edit numbers assigned to predefined messages. When creating custom messages, use only message numbers within the 10,000,000 to 10,999,999 range.

See: About Message Names

See: About Message Numbers

- The translation notes for predefined messages might contain internal content that you can disregard.

See: About Translation Notes

- The message category, severity, and logging enabled option are related to the incident and logging process.

See: About Grouping Messages by Category and Severity

See: Understanding Incidents and Diagnostic Logs with Message Dictionary

Message Text and Tokens

- The message text comprises various components, some of which are displayed only to select users. To determine which component of the message text is displayed to a particular user, set the Message Mode profile option (FND_MESSAGE_MODE) at the user level for that user. The message component short text is visible to all users and therefore, the profile option does not apply to this component. Also, the profile option applies only to messages in the message dictionary.

See: About Message Components

- Tokens are variables that represent values to be displayed in the message text.

See: About Tokens

Profile Options and Related General Preferences: How They Work Together

Some Oracle Fusion Middleware Extensions for Applications profile options are related to general preferences in the global area.

Preferences

The related general preferences are Default Application Language, Territory, Date Format, Time Format, Currency, and Time Zone. When the user changes any of these preferences, the stored values in LDAP are updated accordingly.

Profile Options

The corresponding profile options are Default Language, Default Territory, Default Date Format, Default Time Format, Default Currency, and Default User

Time Zone. No matter what you set for these profile options at any level, the preferences settings, or LDAP values, take precedence. The profile option value is used only if the LDAP value is not available. Updating the profile option value does not automatically update the value in LDAP or preferences.

Define WebLogic Communication Services Configuration

Click-to-Dial: Top Tasks

Click-to-Dial is a feature of the customer contact process. Phone communication to customers and employees is initiated with a click of the mouse, leveraging your customer contact information and the application context. The Click-to-Dial feature uses Oracle WebLogic Communication Services, OWLCS, to enable communications. Applications that provide the Click-to-Dial functionality do so primarily through contextual actions in the application.

Additionally, Click-to-Dial utilizes Oracle Fusion Interactions as an optional transaction logging feature that will track information about the call such as the customer, call participants, a timestamp noting the start time of the call, the direction of the communication, in or outbound, and the resolution code.

Click-to-Dial integrates with your telephony environment and must be manually enabled in your deployment. This topic highlights what is required to set up the Click-to-Dial feature and to implement logging of the calls made using the Click-to-Dial feature.

Terms used in setting up these communications

- PSTN: Public switched telephone network is the network of the world's public circuit-switched telephone networks.
- SIP: Session initiation protocol, an open signaling protocol standard that is used to set up phone calls
- TPCC: Third Party Call Control enables an application to control the telephony network to set up calls automatically.
- OWLCS: Oracle WebLogic Communication Services. Offers the TPCC service to Fusion applications and sets up the calls via SIP integration with the telephony network.

The set up task list Define WebLogic Communication Services Configuration delineates four tasks required for the correct configuration and implementation of Click-to-Dial. There is an optional task, separate from the set up task list, required for implementing Interaction logging.

Information about implementing Click-to-Dial can be found in the Oracle Fusion Applications Administrator's Guide. Detailed information about configuring and maintaining WebLogic Communication Services is found in the Oracle WebLogic Communication Services Administrator's Guide

Configure and Deploy WebLogic Server

- Deploy WebLogic Communication Services: After the Oracle WebLogic communication server is deployed, this manual task activates the server.

See: Oracle WebLogic Communication Services Administrator's Guide

Integrate Communications Services

- Integrate WebLogic Communication Services with Telephony Network: This manual task integrates communications within the telephony environment. OWLCS must be configured to interface with the specific characteristics of the telephony network.

See: Managing Oracle WebLogic Communication Services for Click-to-Dial Functionality

Specify the Domain and Address

- Register a URL for the telephony gateway or soft switch for SIP domain: This task defines the Server protocol, defaulted to http, the external server host address and external server port address. The Associated Modules section is not required for setup. You can also perform this as a manual task using the Oracle Fusion Topology Manager to configure the address of the SIP Public Switched Telephone Network (PSTN) gateway or SIP soft switch serving the users within that domain. This address is needed by Click-to-Dial to correctly form the SIP addresses required by WebLogic Communication Services. See the link to [Configuring PSTN Gateway Address Using Topology Manager: Worked Example](#).

Enable Click-to-Dial

- After configuring the server and defining the SIP domain, perform the Enable Click-to-Dial task. This task sets the value of the profile option Enable Click-to-Dial to 'Yes.'

Call Logging via Fusion Interactions

- To initiate the Interaction based logging for Click-to-Dial, set the profile option Call Interaction Logging Enabled to 'YES.'

Enhanced Click-to-Dial Setup: Explained

Click-to-Dial is the feature that enables the user to initiate outgoing calls via hyperlinks. Extended Click-to-Dial enhances the click-to-dial experience by adding an agent communication toolbar. The communication toolbar is a real time dashboard for the agent's voice communications. It displays the agent's telephony state, current status of a call (during call setup and actual conversation) and provides a set of voice controls as well as ability to access and modify the current interaction. This toolbar displays when a call is made or

received, allows the agent to transfer the call or consult with another agent and return to the original call.

When a call is received, Enhanced Click-to-Dial displays the identity of the inbound caller via a reverse directory search and can open an interaction page for logging the call. If a search fails for the contact name, the company may be identified and a search dialog will be displayed that enables you to search for your contact to initiate the interaction logging. If the company is not identified, you can still search all contacts to generate the interaction. You can manually call up the interaction dialog at anytime from the toolbar.

When a call is complete, the application can display a wrap up interaction dialog to record the results of the call.

The communications toolbar and the communication services provided use a slightly different infrastructure than previously delivered. To enable this infrastructure there are four tasks:

- Manage Click-to-Dial Profile Options
- Manage Communications Connectors
- Manage Agent Connector Settings
- Manage Contextual Navigation Rules

Manage Click-to-Dial Profile Options

This is the first task for setting up the Oracle Fusion Enhanced Click-to-Dial feature.

These profile options determine if the feature is available, whether individuals have access to the toolbar, and the in and outbound logging preferences.

User Profile Option Name	Description
Enable Enhanced Click-to-Dial	Enables the site wide Enhanced Click-to-dial functionality. It is recommended that this profile option be only enabled after completing associated setup tasks for Click to dial.
Enable Click to Dial Toolbar	Indicates whether the Click to Dial toolbar should be displayed for a user with a click-to-dial enabled job role. It is defaulted at the Site level to Yes. Individual users can be opted out of the toolbar display by adding a User level entry with the profile value set to No.
Automatic Inbound Interaction Logging	When set to Yes, interactions are automatically logged for inbound calls. If set to No, no wrap up dialog will be displayed at the end of a call.
Inbound Interaction Logging Type	When "Automatic Inbound Interaction Logging" is set to Yes, controls how interactions are logged for inbound calls. Can be set for All Communications or by the Party Usage Filter Rules, for example: all eligible customers. If left blank an interaction is created for all inbound calls.
Automatic Outbound Interaction Logging	When set to Yes, interactions are automatically logged for outbound calls. If set to No, no wrap up dialog will be displayed at the end of a call.

Outbound Interaction Logging Type	When "Automatic Outbound Interaction Logging" is set to Yes, controls how interactions are logged for outbound calls. Can be set for All Communications or by the Party Usage Filter Rules, or example: all eligible customers. If left blank an interaction is created for all outbound calls.
Enable Inbound Screen Pop	If set to Yes, the contextual navigation for inbound call is performed unless the toolbar option Contextual Navigation is set to no. If set to No, no screen pop or interaction dialog is displayed.
Enable Outbound Screen Pop	If set to Yes, the interaction dialog for outbound calls is displayed when an interaction is automatically logged, unless the toolbar option Contextual Navigation is set to no. If this profile option is set to No, the interaction dialog is not displayed.
Click to Dial Toolbar Auto Login	Click-to-Dial automatic log in is not used at this time.

Manage Communications Connectors

The communications connectors provide the parameter settings required to establish connectivity between the Fusion application and a telephony system. The Oracle Contact on Demand connector, and a software development kit (SDK) for developing additional connectors are delivered.

The Manage Communications Connectors user interface enables you to add other connectors and connector types as required by your specific communications needs.

Define the connector types before designating the connectors and parameters. You can add Agent specific connector parameters such as their agent ID or extension number. These parameters are passed to the connector and used as needed to communicate with the telephony system.

Define the connectors. There are four parameters that are provided to pass location information for the various downloads and servers. A fifth parameter is available to pass any connector specific parameters to the telephony system.

- JavaScript Location - The location from where 3rd party JavaScript will be downloaded. The JavaScript will be run within the browser and can be either a transport library implementation for communication to a vendor specific connector or the actual connector itself. If left blank, Oracle's default transport library will be used.
- Runtime Location - The connectors runtime location, an internet protocol (IP) address or domain name for the connector or URL. This is used by Fusion Click to Dial toolbar to communicate with a connector that runs outside the browser.
- Download URL - The location /URL from where a connector will be downloaded and is reserved for the use of the Oracle Contact On Demand Connector.
- Server Location - The telephony system server runtime location, an IP address or domain name.

- Parameter String - Additional connector parameters for the specific telephony system in addition to the parameters above. This is reserved for the specific use of the connector.

Manage Agent Connector Settings

An Agent is required to sign in to a connector to establish the connection to the telephony system to make and receive calls. This task establishes the relationship between the connector, the agent, the agent group, identifier and password.

After adding a new row to the connector settings, you can search for the agent name using the orange glyph that opens a search dialog. The assignment of the agent groups, an agent ID, and password are usually defined and entered by the administrator. An agent can also self-provision themselves through the agent sign in dialog box. This dialog box is not available through the setup UI. It can be launched in the transaction flow from the toolbar UI. On the sign in dialog box, click on edit connector. User can then create a connector for himself and also provide additional agent parameters for the connector. These data values are used exclusively for an agent signing in to a connector. There is also a sub-page for entering any other parameters specific to that agent that are required by the third party telephony system and not covered in the delivered configuration parameters. Some examples required for Oracle Contact On Demand Connector are the `AcidUserID` and `AcdExtension`. These two parameters are used in the agent transfer scenario where a list of agents together with the Extension will be available for the user to choose to which agent the call will be transferred. `AcidUserID` is a special id that Oracle Contact On Demand (COD) assigns to each COD user. `AcdExtension` is the virtual extension that COD assigns to each user. Therefore, instead of knowing the exact number to dial in order to reach out to an agent, the extension can be used. COD determines the phone number to call from that virtual extension. The agent connector parameters will differ from one connector to another.

Note

An Agent also has the ability to set this information through the Connect button in the toolbar. Therefore, the Manage Agent Connector Settings task can be skipped and delegated to each agent if so desired.

Manage Contextual Navigation Rules

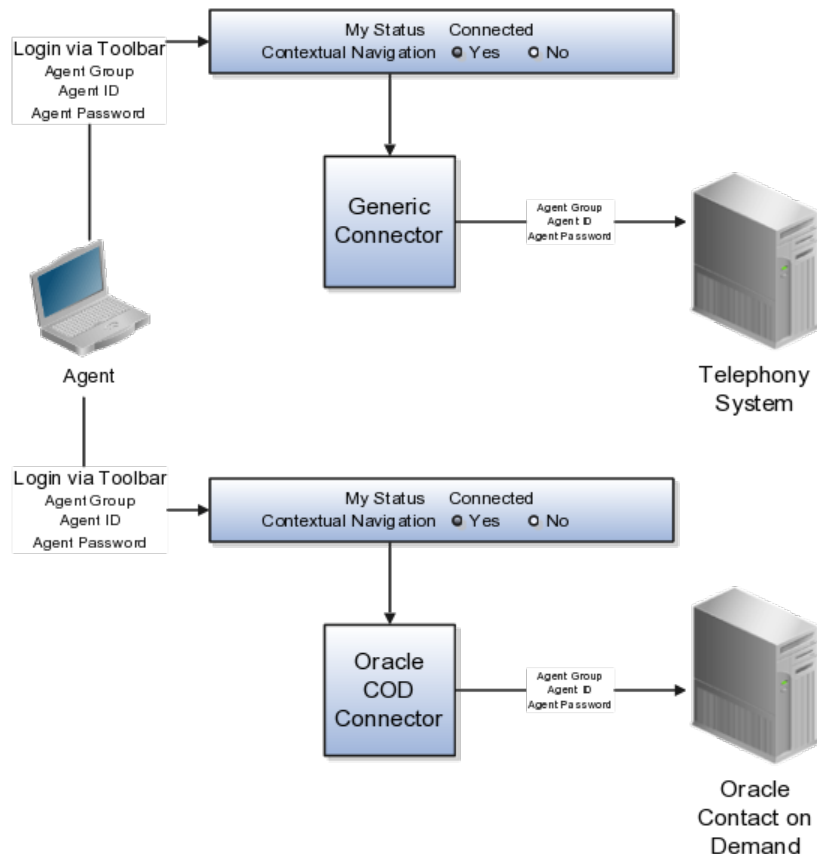
Upon receiving an incoming call, the connector will follow a set of rules to determine which UI to open upon connecting. This feature can be turned on or off on the Click-to-Dial toolbar by choosing the Yes or No radio button for Contextual Navigation.

Navigation rules are organized in Rule Sets. There can only be one active contextual Navigation rule set at a time. The navigation rules themselves are comprised of a **Call Data Token** and the **Target Navigation Page**. One way the call data token is set is by using the values returned by Automatic Number Identification (ANI) and the Dialed Number Identifier Service (DNIS) from the telephony system, in combination with IVR data provided by the caller such as Organization ID or account number. Call data token value or a default value are used as the value for task flow parameter in order to launch the target navigation page when the connection is established.

The Target Navigation Page is the page that will open upon accepting the call. Target Navigation Pages are defined by a name and a task flow identifier. Each navigation page, or task flow, can have one or more parameters. The parameter can be set with either a default value or set with a call data token so that a value is set at runtime using the call data token name.

Enhanced Click-to-Dial Components: How They Work Together

Oracle Fusion Click-to-Dial is a user friendly way to receive calls from your customers and contacts and to record those interactions. Outgoing calls are made through the hyper-linked phone numbers seen displayed with customer and contact information and through contextual action dialogs. A Click-to-Dial tool bar is used as a dashboard for call status and taking actions on inbound and outbound calls. The tool bar is supported by an infrastructure of connectors and telephony systems that enable this to happen. A delivered generic application program interface (API) enables Click-to-Dial to work with any switch or telephony vendor. Oracle does not author or maintain these connectors, except for Oracle Contact Center On Demand (COD), which is also a reference implementation. For this release, the COD connector is supported only on Windows desktops and Linux. This restriction will be removed in later releases.



Agents

An agent initiates their availability for inbound voice communications by signing in to a connector via the tool bar. The agent signs in via a single connector for that toolbar session. Each connector is associated with an agent or agents. The agent has an assigned agent group, agent identifier and password for each connector. This enables the Agent to sign on to a connector and use the tool bar to receive and transfer calls to other agents and customers or contacts. There may also be other parameters required at this agent level for the telephony system. These can be designated at the agent connector parameter level and passed by the connector to the telephony system. Agents may also record each of these transactions via the Oracle Fusion Interaction features.

Connectors

Connectors perform the linking of the agent to the telephony system that enables the outgoing and incoming calls via Click-to-Dial. The Oracle Contact on Demand Connector is delivered. Contact On Demand is a multichannel, all-in-one call center application that itself connects to the Public Switched Telephony Network (PSTN) to provide voice communication capability. Other connectors may be added using the delivered template.

Telephony Systems

These are third party systems used to establish telephone communications to and from the customer site. They can operate over the internet protocol layer or on hard wired telephony networks.

Tool Bar

This user interface to the Click to Dial feature enables you to establish the connection with the software connector to the telephony system. You may receive, ignore, hold, or transfer a call from your customers and contacts. You can also consult with another agent with the calling party on the line or on hold.

The Oracle Fusion Interactions application can be used to record the particulars of the call using the interactions user interface. Upon receiving a call the tool bar will use the incoming phone number to identify the caller. If the caller is known, an interaction window will be created and opened. If the caller is unknown, the tool bar will popup a window allowing you to search the customers or contacts available to you. If the company is known but not the caller, the list of contacts available to you from that company will be searched. You can also manually open a search dialog. Once the call is completed the Interaction wrap-up screen can be programmed to popup at the end of the call enabling the Agent to finish noting any further call details.

Define Common Order Orchestration Configuration

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.

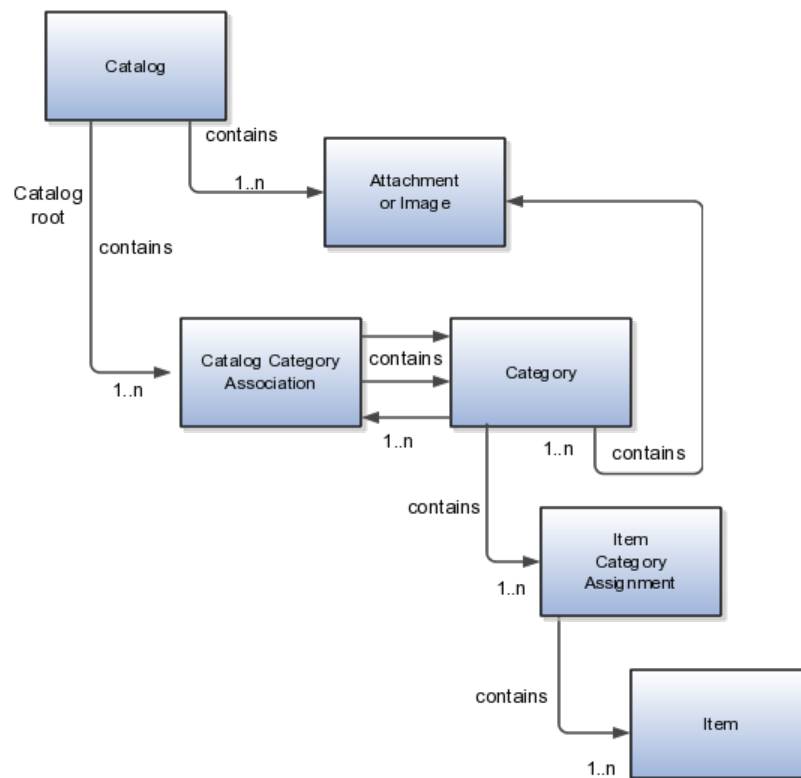
Define Catalogs: Create Catalogs

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.



Catalog

A catalog is a collection of categories that are organized to define a classification of items. The top most level of a catalog is the catalog root. All categories for the 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.

Catalog Formatting: Explained

The format of a catalog is defined at the time the catalog is created and controls the behavior of the catalog at runtime.

When you format a catalog the layout controls three main areas and includes the following tasks, some fields are required, and others are optional.

- Catalog configuration
- Date enablement
- Category sharing

Catalog Configuration

You can configure the catalog, and this affects how the content behaves. The catalog configuration contains a set of attributes that define the catalog configuration. These attributes interact to define the runtime behavior of the catalog.

The configuration functions are:

- **Catalog code:** A unique identifier that is used.
- **Catalog structure:** The key flexfield structure used to define the catalog.
- **Controlled at:** Controls how items are assigned to categories and has two values. The first value is master level, which enables the automatic assignment of items to all child organizations associated with the master organization, if the current context is a master organization. The second value is organization level, which assigns the item only to the organization in the current context.
- **Default category:** Applies any time a new item is created. The newly created item is assigned to this category within the catalog automatically. The automatic assigned is controlled by the functional area.
- **Catalog content:** Controls what content can be added to the catalog and where the content can be added. This attribute has three values:
 - **The Item at leaf levels** allows items to be added only to the bottom level categories in the hierarchy.
 - **The Items at all levels** allows items to be assigned to any category in the hierarchy regardless of level.
 - **Categories only** allows categories to be added only to the catalog.
- **Allow multiple item category assignment:** When this option is selected, you can assign an item to one or more categories in the catalog. The default is deselected, which means that each item can be assigned to only one category in the catalog.
- **Enable hierarchies for categories:** When this option is selected, you can create a hierarchy for the catalog. The default is deselected, which means that the catalog cannot have a hierarchy and categories are associated with the catalog root.
- **Enable automatic assignment of categories:** When this option is selected, the catalog is built by automatically associating all categories, based on matching the catalog structure value to the category structure value.

Catalog Date Enablement

The date enablement function controls when the catalog is in an active state or inactive state by using the start date and end date attributes.

Category Sharing

The category sharing function enables sharing by reference to categories from a designated source catalog.

The sharing function has these attributes:

- Source catalog: A catalog that does not have sharing enabled from which categories, category hierarchies, and assigned items can be added to the catalog.
- Sharing content: Controls what content can be added from the source catalog. This attribute has three values:
 - Categories only: Only categories without assigned items can be shared.
 - Items only: Only categories with assigned items can be shared.
 - Items and categories: All categories can be shared.

Catalog Details: Explained

You can change a default category so that you can use it for item creation, or modify the inactive date so that the category is no longer used as you update a catalog. You can correct mistakes or reclassify the category due to shifting relationships within the category hierarchy.

You can view and edit a catalog on the Edit Catalog page when you have editing rights. For users that do not have rights to edit, the page is in read only mode.

The following aspects are important regarding managing and editing catalog details:

- Catalog header region
- Catalog detail tab
- Category hierarchy tab

Catalog Header Region

This region contains the catalog name and description, the selection of the default category and the start and end date for the catalog.

Catalog Detail Tab

The Detail tab contains:

- The configuration attributes for the catalog that controls the runtime behavior for the catalog.
- The sharing attributes for the catalog which controls the source catalog that will be used for sharing from and what content can be shared.
- The additional information which contains the descriptive flexfields defined for the catalog.

Category Hierarchy Tab

This contains the category hierarchy region in which the category hierarchy can be created and maintained. In addition, items can be assigned, and the usage of

the category in other catalog can be viewed, and the attributes for the category and catalog category association can be edited.

Automatic Assignment Catalogs: Explained

The automatic assignment catalog feature enables you to reduce the cost of creating and maintaining a catalog. It is a simple way to create a nonhierarchical catalog because you do not have to add categories manually to the catalog.

All categories that have the same category structure value as the catalog are automatically assigned and associated to the catalog when you create a catalog category association for each category. Note that if you create a category in another catalog with the same structure value as the automatic assignment catalog, the category is added to your catalog. The categories displayed for auto assignment catalogs are refreshed only at startup and after you save.

Automatic Assignments

The automatic assignment feature is enabled during catalog creation when you select the **Enable automatic assignment of category** check box. When you open a new catalog, any categories that have the same category structure value as the catalog structure value for the catalog are automatically assigned to the catalog.

For example, Purchasing may maintain a master catalog containing all categories that represent commodities. Each commodity team can create categories for their commodity in their own catalog.

- The master catalog for Purchasing is named Purchasing and is configured during creation to support the automatic assignment of categories.
- The Electronic commodity team creates a catalog named Electronics and proceeds to create categories that represent the classification of their commodity. The Electronic commodity team creates the categories televisions, computers, and home theaters.
- The other commodity teams create and maintain separate catalogs.
- Because you enabled automatic assignments for the Purchasing catalog, any categories created by the commodity teams are added to the catalog automatically. The Purchasing managers can view the collection of all commodities represented as categories in the Purchasing catalog.

Define Catalogs: Manage Catalogs

Catalog Edits: Explained

The Edit Catalog dialog is a shared page that has two modes, view and update. The view mode displays the selected catalog in a read-only file. The update mode displays the selected catalog in an editable file. You must have edit catalog

privileges to access the catalog in update mode. You can edit only an active or future-dated catalog.

The following fields are editable in the catalog:

- **Catalog Name**
- **Description**
- **Start Date**
- **End Date**
- **Default Category**
- **Allow multiple item category assignment**
- **Addition Information**
- **Category Hierarchy**
- **Category Details**
- **Items assigned to category**

Default Category

You can edit this field to select another category as the default category for item creation. You cannot remove the default category if the catalog is assigned to a functional area that requires a default category to be specified.

Allow Multiple Item Category Assignment

This check box is editable only until you assign an item to a category in the catalog.

Addition Information

You can edit the values of the descriptive flexfields attributes.

After you make changes, clicking the **Save** button saves the changes to the database but will does not close the Edit Catalog page. Clicking the **Save and Close** button saves the changes to the database and closes the Edit Catalog page.

Categories and Catalog Relationships: Explained

Catalogs are used to organize and classify collections of items by associating categories to the catalog. The categories are organized to form a taxonomy and items are assigned to the categories. When a category is associated with the catalog a catalog category association is created which specifies the relationship of the association. The catalog category association may also represent the relationship between two categories, for example a relationship between a parent category and a child category.

The following aspect is important regarding catalog category association:

- Date enablement attribute value

Catalog Category Association

The catalog category association is date enabled providing the control of when the catalog category association is active in the catalog and when the catalog category association is inactive. The catalog category association has two attributes to support date enablement; the start date and the end date. The start date is value is the first day that the catalog category association is available or active for use and the end date is the last day the catalog category association can be used, after this date the catalog category association is inactive. The date enablement attribute values are also used to control the visibility of content and the behavior of the category in the catalog. If a category association is inactive or end dated, having the value of the end date attribute past the current date, then the items cannot be assigned to the category.

A catalog category association will be set to inactive state when the category referenced by the catalog category association is set to an inactive state automatically, but the display will not be refreshed automatically.

Date Enablement for Catalogs and Categories: Explained

The catalog, categories, and catalog category association use date enablement to determine if the object specified is active or inactive based on the start date and end date. The following are date enablement definitions:

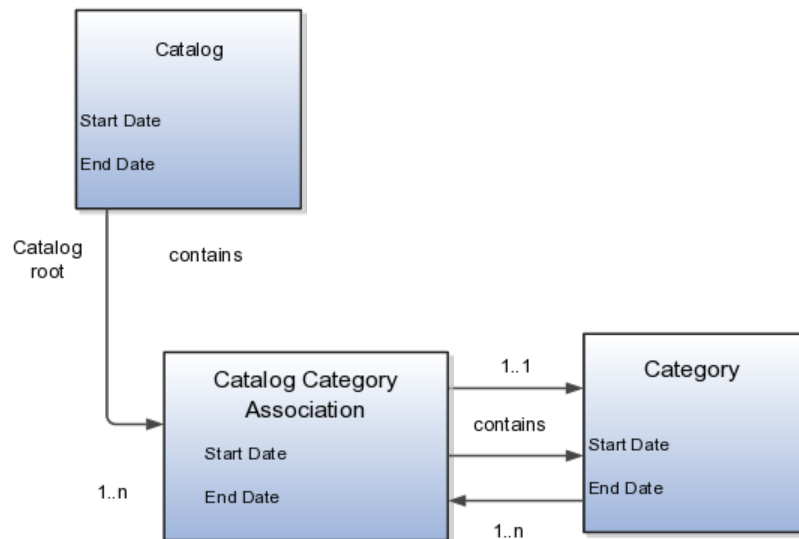
- **Active** An object is active when the current date is later than or equal to the value of the start date, but earlier than or equal to value of the end date.
- **Inactive** An object is inactive when the current date is later than the value of the end date.
- **Future dated** An object is future dated when the current date is earlier than the value of the start date.

You set the date enablement attributes are used to determine when a catalog, category, or catalog category association is used or visible.

- On the Manage Catalog page, a table filter determines which catalogs appear. The default value for the choice list is **Active**, indicating that only active catalogs will be displayed. You can select the value **All** to view both active and inactive catalogs.
- On the Edit Catalog page, on the category hierarchy tab, two table filters determine what categories and catalog category associations appear. The default values for the two choice lists are **Active**, indicating that only active categories and active catalog category associations will be displayed. You can select the value **All** to view both active and inactive categories and catalog categories associations.

- Other applications also use the date enablement attributes to filter information retrieved through application programming interfaces or services for catalogs.

The following illustration provides the date enablement attributes for these objects. The catalog, category, or the catalog category association has an internal state that is active or inactive.



The following aspects are important regarding date enablement for catalogs and categories:

- Start date
- End date
- Catalog and category objects
- Catalog category association
- Catalog and category rules

Start Date

The start date is defined as the first date that the object can be active. The start date can be future dated by setting the value to a date later than the current

date. The start date value defaults to the system date if no date is entered during catalog or category creation.

End Date

The end date is defined as the last date that the object can be active. The object is end dated one second after the date specified by the value of **End Date**, that is the next day at 12:00:01 a.m. You cannot set the end date in the past. Also, you can change the end date from a condition when the object is ended to a new end date greater than or equal to the system date, causing the object to go from inactive to active. The end date value is optional during catalog or category creation.

Catalog and Category Objects

The start and end dates have been added for the catalog and catalog category association. The inactive date for categories has been renamed as the end date and the start date has been added.

Catalog Category Association

The catalog category association is used to specify the parent and child relationships between catalogs and categories and for category to category relationships. The catalog category association date enablement is independent of the category data enablement, except for the case where the category is end dated; the association is ended automatically as well. The catalog category association dates represents the state of the category for the catalog in which the category is associated.

Catalog and Category Rules

When a catalog is inactive the following rules apply:

- All operations for the catalog are disabled; the catalog is not editable.
- The catalog cannot be used in other processes.
- The catalog can be viewed only if you set filters on the Manage Catalog page to a value of **All**, enabling you to view active and inactive catalogs.

When a category is inactive the following rules apply:

- All operations for the category are disabled; the category is not editable.
- The category cannot be added to other catalogs.
- The category can be viewed only if you set the filters on the Edit Catalog page to a value of **All**, enabling you to view active and inactive catalogs.
- The system sets the catalog category association for the inactive category to inactive.

When a catalog category association is inactive the following rules apply:

- The category may be inactive or active; if the category is active it can be edited.

- The catalog category associations and related category can be viewed only if you set the association filter on the Edit Catalog page to a value of **All**, enabling you to view active and inactive catalogs.

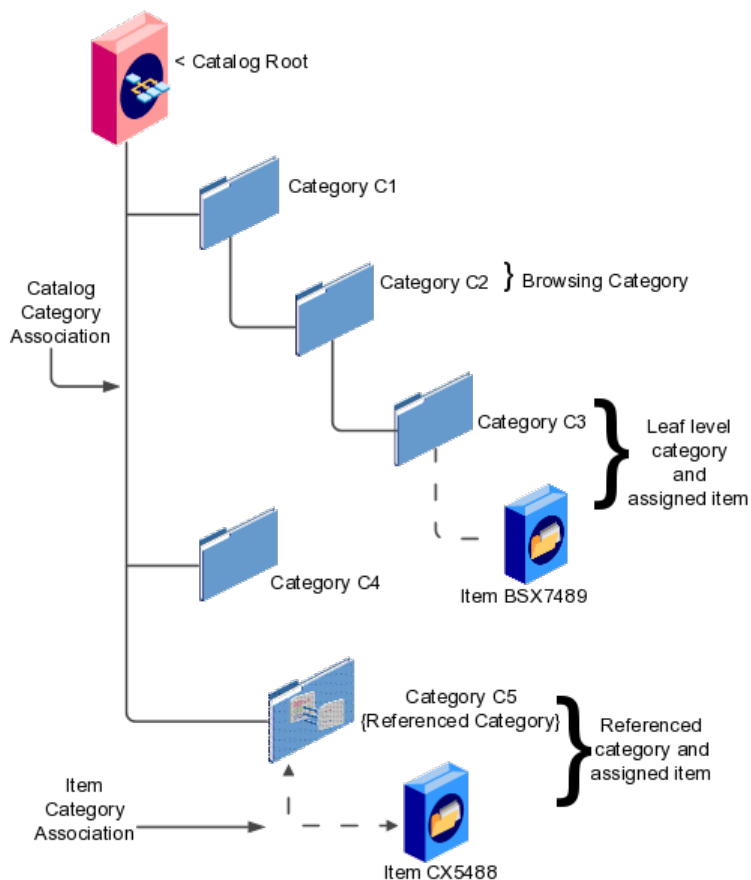
When a catalog is future dated the following rules apply:

- All the operations of the catalog are enabled and the catalog is editable.
- The catalog can be used in other processes, if allowed.
- The catalog can be viewed only if the you set the filters on the Manage Catalog page to value of **All**.

Catalog Hierarchies: How They Fit Together

You use catalogs to organize and classify collections of items by associating categories with the catalog. You organize the categories to form a taxonomy and assign items to the categories. When you associate a category with the catalog, a catalog category association is created which specifies the relationship of the association. The catalog category association may also represent the relationship between two categories, for example, a relationship between a parent category and a child category.

The following diagram shows the relationships of the category hierarchy components:



Components

The components of a category hierarchy are:

- **Catalog root:** The topmost node in category hierarchy that represents the object called catalog.
- **Category:** The catalog component that is used to represent the classification structure.
- **Catalog category association:** The line in the diagram represents the relationship between a catalog and category or between a parent category and child category.
- **Item category assignment:** The dotted line in the dialog represents the relationship between a category and an item.
- **Reference category:** The category C5 in this diagram is shared as a reference category from a source catalog.
- **Leaf level category:** The lowest or bottom-level category in a category hierarchy. You can assign items to all levels in a category hierarchy if you configure the catalog to support this.
- **Browsing category:** The category C2 in this diagram is a browsing category. Browsing categories are categories that you add to the category hierarchy for the purpose of classification and do not have items assigned to them.

The category hierarchy does not have a limit on how many levels can be represented. The category hierarchy can have multiple hierarchies within a single category hierarchy.

Category Edits: Explained

Categories can be edited only from within an Edit Catalog page, the category hierarchy tab. The category can be edited by selecting row for the category in the category hierarchy table and editing the category information in the category detail panel. The category can only be edited if the category is active and the catalog is active or future dated.

The category information can be edited in both the details and items tabs.

Details and Items Tabs

The following fields are editable in the category:

- **Category name**
- **Description**
- **Attachments**
- **Category start date**
- **Category end date**

- **Items assigned to category**

After changes are made the **Save** button will save the changes to the database but will not close the Edit Catalog page. The **Save and Close** button will save the changes to the database and close the Edit Catalog page.

Catalog Category Association: Explained

The catalog category association can be edited only within the Edit Catalog page, in the category hierarchy tab. The catalog category association start date and end date attributes can be edited in the details region.

Category Catalog Associations

You select the category in the category hierarchy table for the catalog category association that is being edited, the category details are displayed in the right hand panel. The association start date and association end date are the only editable fields.

After completing the edits, click on the **Save** button to save your changes to the database, the Edit Catalog page will not close. The **Save and Close** button will save the changes to the database and close the Edit Catalog page.

Category Details: Explained

You can update category details when you select the row with the category in the category hierarchy table, the category details are displayed in the right hand panel in the user interface in an edit mode for all native categories. The category detail region contains information about the category that is associated to the catalog. It also contains the association start and end dates.

You can view and edit a catalog on the category details tab when you have editing rights. For users that do not have rights to edit, the page is in read only mode.

The following aspects are important regarding managing and editing category details:

- Category details tab
- Items tab
- Where used tab

Category Details Tab

The details tab contains information about the category that has been associated to the catalog. This information appears in all catalogs, since a category can be associated to one or more catalogs. The details tab contains the category configuration, category date enablement, association date enablement, and the additional attributes for the category.

The details tab contains attributes that define a category. Unstructured information is added through attachments. Images are added to a category and are displayed in the category details tab.

Items Tab

The item assignments are specific to the catalog where the category is associated.

Where Used Tab

The Where used tab contains a list of catalogs that the category is associated with.

Categories: Explained

You can create categories only in the context of a catalog, on the Edit Catalog page, Category hierarchy tab. When you select the **Create** icon in the category hierarchy table, it launches the Create Category dialog.

Consider the following important aspects when creating categories for catalogs:

- Create category region
- Configuration region
- Date enablement region
- Additional information region

Create Category Region

Enter a name and a meaningful description of the category in the create category region. Optionally, you can add an image and an attachment to this category.

Configuration Region

The key flexfield is determined during creation based on the catalog structure of the catalog. Enter the key flexfield segment values for the category. The number of key flexfield segment values depends on how you define the key flexfield at setup time. The category structure is the key flexfield structure instance that you create as part of the setup. When you define the key flexfield structure instance, you define the segments for the structure instance. For example, the family group and class group are segments. The segments appear in the Create Category dialog based on the key flexfield structure instance that you select.

The default value of the category content selection value is **Items and Categories**, but you can change the value. The values in the category content choice list vary based on the catalog content value.

The category content attribute value controls the content that you can add to this category.

- **Items Only:** Select to add only items to the category

- **Categories Only:** Select to add only categories to the category
- **Items and Categories:** Select to add both items and categories to the category

Date Enablement Region

Date enablement determines if an object is active or inactive based on the start date and end date. When categories are created, the default start date value is the current date. You can move the category start date beyond the current date to a future date within the category. The end date value is optional.

Additional Information Region

The additional information region contains all descriptive flexfield attributes that you set up for categories. You can edit the values of the descriptive flexfield attributes at the time of category creation.

After you complete the required fields for the catalog, clicking **OK** creates the category in the database, adds the category to the point of selection in the category hierarchy, and closes the dialog.

Category Moves: Explained

You use the move category function in the category tree table region of the Edit Catalog page. This is a table row action. The dialog is launched when you select an active or future dated category within the catalog and select this action. The move category function is disabled when the **Enable hierarchies for categories** check box is not checked or left unchecked.

Consider the following important aspects when moving categories within catalogs:

- Identifying the new parent

Identifying the New Parent

The dialog provides the current category parent and allows you to pick a new category parent. Only the legal category parents are displayed in the choice list.

The category list within the **New Parent** choice list is filtered by based on a set of rules:

- The new parent category must be an active or future dated category; the end date value of the category must be later than the current system date.
- The value of the category content for the new parent category must allow the selected category to be added; the legal values are items and categories and categories only.
- A selected category associated with the catalog at a level below the categories at the root categories can be moved to the root of the catalog.

- The new parent category catalog category association must be active; the end date value of the catalog category association must be later than the current system date.

Import Category Hierarchies: Explained

Category hierarchy can be created and maintained through a spreadsheet interface reducing the amount of time to create and maintain catalogs. Existing catalog content can be exported and the content used in other catalogs for catalog category hierarchies.

The following aspects are important regarding category hierarchy import used in catalogs:

- Spreadsheet interface
- Export category hierarchy

Spreadsheet Interface

You can manage the catalog category hierarchy to use the spreadsheet interface that is available in the Edit Catalog page by using the **Export Hierarchy** button to download existing catalog content, modify this content in a spreadsheet, and upload the content back into the Product Information Management application.

Export Category Hierarchy

You use export category hierarchy for example, when you need to provide the category hierarchy to a partner. Your partner has the capability to import the catalog file using an Excel spreadsheet.

You can export the category hierarchy from our catalog and it can be used by partners. If your partner has the Oracle Product Information Management solution, they can directly import the category hierarchy into their catalog.

Catalog or Category Attachments: Explained

Catalogs and categories support attachments and use a common component for managing attachment content. You can add attachments on both the Create Catalog and Edit Catalog pages.

The attachment component displays a green plus sign icon indicating that no attachments are available for the object. The Attachment dialog appears when you click the green plus sign icon. You define the attachment by selecting the attachment type, file name or Uniform Resource Locator (URL), title, description, and by indicating whether the attachment can be shared with other objects. Once you define the attachments and click the **OK** button, that attachment title appears in the attachment component region of the page along with a red X icon that you can click to delete the attachment.

The attachment file types are:

- File
- Repository File/Folder
- Text
- URL

File

You must provide a title for the file and create a description for the attachment. You select a file to upload from your desktop.

Repository File/Folder

You click the **Browse** button to attach a repository file/folder from the document repository to a catalog. The attachment repository contains existing attachments and is organized as a set of folders. The **Browse** button launches the Attachment Repository dialog to enable you to select an attachment. You must provide a title for the repository file/folder and create a description for the attachment.

Text

Enter the text string in the field that you want to appear as an attachment. You must provide a title for the text and create a description for the text attachment.

URL

Enter the URL address to a web page that you want to attach to the catalog. You must provide a title for the URL attachment and create a description for it.

The **Share** check box alerts users that you added an attachment and the date that you performed the task.

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 Manage 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.

How can I define category hierarchies?

Categories can be organized to represent classification taxonomies. The hierarchy organizations for categories have parent and child relationships that form a tree structure. The category hierarchy is created and maintained within the Edit Catalog page, category hierarchy tab. The category hierarchy is shown in true relationship to the way it is defined.

The category hierarchy can be created using two methods: the first is manually creating the hierarchy by adding referenced categories, duplicating categories or creating category for the catalog.

The second method for creating the hierarchy is by importing the category hierarchy through the spreadsheet interface. The category hierarchy can be exported from other catalog or other sources, edited and imported into a new catalog, additionally it can be added manually to the spreadsheet.

The category hierarchy can be edited using **Move Category**. The catalog category association cannot be deleted, but can be end dated to make the catalog category association inactive. The category hierarchy table provides a choice list filter that controls what catalog category associations and categories area displayed based on the date enablement. The category hierarchy can also be edited by exporting the complete hierarchy, editing it and importing the category hierarchy back into the catalog.

How can I duplicate categories?

You can select and duplicate a category as a quick way to create a similar category configuration. Selecting the **Duplicate** icon action launches a Create Category dialog that has attribute fields populated based on the selected category attribute values. The category name is prefixed with **Copy_** followed by the name of the selected category. You fill in the required field information in the

key flexfield segment values which are blank. Once the category attributes are updated and the key flexfield segments values are entered, the **OK** button adds the newly created category into the category hierarchy of the selected category you have configured.

How can I add categories?

Categories are catalog components that are associated to a catalog for purpose of classification of items. You can add existing categories to the point of selection which can be a category in the hierarchy or the root of the catalog. If no category is selected, the default is the root of the catalog.

You can add categories by selecting the **Add Category** field and selecting the value **Add Category**. You can then search for existing categories based on the value of the catalog structure for the catalog. You can narrow the search for existing categories by using the **Advance Search** region in the dialog. You can add each selected category by selecting the **Apply** button and the add category region remains open. The **OK** button adds a category if a category is selected and then closes the dialog.

How can I add shared categories?

Adding a shared category is similar to adding an existing category except the category is selected from the catalog that has been designated as a source catalog. The sharing content attribute value determines what content is shared from the source catalog. A category within a source catalog that has been added to a native catalog is also known as a referenced category. You use the drop list menu from the Add Categories menu, and the Shared Category option will be disabled if the catalog has not been configured for category sharing.

How can I add images to a catalog or category?

You can attach an image from your desktop or from a configured repository to a catalog or a category, or both. The image is displayed in the catalog detail and the category detail section of the catalog page. Only one image can be associated with a catalog or category. To attach an image, select the green plus icon to launch the Manage Attachment dialog. The image attachment type can have values of **File** or **Repository File/Folder** and is selected in this dialog. The title you provide for the image attachment will appear under the image that is displayed in the catalog. The description you provide is not displayed. **Browse** will allow you to select the file to be used as the image for the catalog or category. After the information is entered in to the dialog, you click the **OK** button to load the image and the image attachment title will be displayed under the image. The image will not initially be displayed until the catalog is saved. The image can be replaced with another image by selecting the red X to delete the existing image and entering a new image.

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.

FAQs for Manage Default Catalogs

How can I map default catalogs?

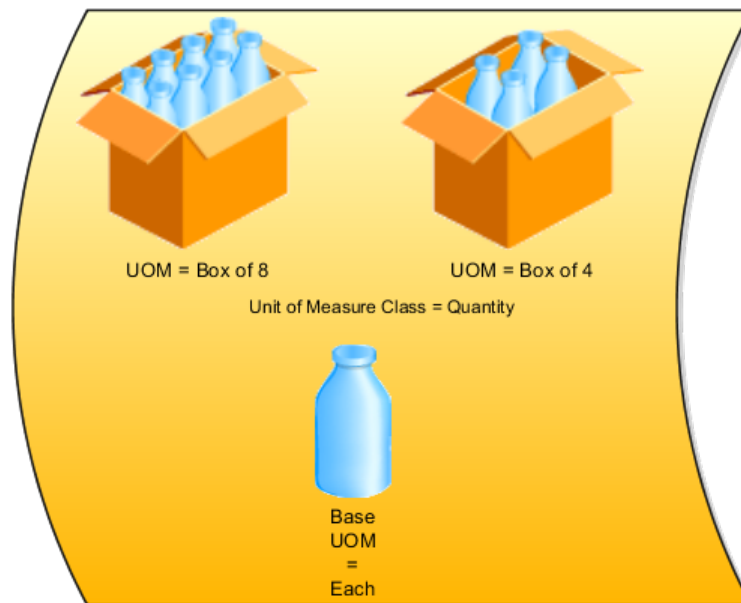
You can map a catalog to be assigned to a functional area such as Purchasing. When a catalog is assigned to a functional area, the catalog will behave based on the rules you defined for that functional area. Only one catalog can be assigned to a functional area.

Manage Units of Measure

Units of Measure, Unit of Measure Classes, and Base Units of Measure: How They Fit Together

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.

Unit of Measure Class	Units of Measure	Base Unit of Measure
Quantity	dozen box each	each
Weight	pound kilogram gram	gram
Time	hour minute second	second
Volume	cubic feet cubic centimeters cubic inches	cubic inches

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.

Unit of Measure Class	Unit of Measure	Base Unit of Measure	Conversion Factor
Quantity	dozen	each	12 (1 dozen = 12 each)
Weight	pound	gram	454 (1 pound = 454 grams)
Time	minute	second	60 (1 minute = 60 seconds)

FAQs for Units of Measure

What's a unit of measure standard conversion?

A unit of measure standard conversion defines the conversion factor by which the unit of measure is equivalent to the base unit of measure that you defined for the unit of measure class. Defining a unit of measure standard conversion allows you to perform transactions in units other than the primary unit of measure of the item being transacted. The standard unit of measure conversion is used for an item if an item-specific unit of measure conversion has not been defined.

What's a UOM interclass conversion?

A UOM interclass conversion defines the conversion between the source base unit of measure ("From Base UOM") in one unit of measure class ("From Class") and the destination base unit of measure ("To Base UOM") in a different unit of measure class ("To Class").

For example, the item is gasoline. The From Base UOM (of the From Class called "volume") is liters. The To Base UOM (of the To Class called "quantity") is Barrels. The conversion is 158.76 liters (volume) to 1 barrel of oil (quantity).

What's a UOM intraclass conversion?

A UOM intraclass conversion specifies the conversion between a unit of measure (the "From UOM") and the base unit of measure of the same class.

For example, the item is soda pop. The unit of measure class is Quantity. The From UOM is Case (CS). The base unit of measure is Each (EA). The conversion is 24, to specify that 1 CS = 24 EA.

Manage Customers: Define Source Systems

Source Systems: Explained

You can set up source systems to enable users to identify the source of the data they are importing into the Oracle Fusion database. 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 trading community member 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 Fusion Applications database IDs. You can create a source system name and description to provide information that is more descriptive than the source system code.

Note

You cannot update the source system code once you have created the source system.

Source System Type

You must set up a source system as either a Spoke system, such as a legacy system, or a Purchased system, such as data from Dun & Bradstreet.

Enable for Items, Trading Community Members, Order Orchestration and Planning, and Assets

You should select which types of entities will be imported from the source system into the Oracle Fusion Applications database from the following:

- Items
- Trading Community Members
- Order Orchestration and Planning
- Assets

You can select one or more of these entity types as required for the source system. It is important to enable the correct entity types because each import UI filters source systems based on their entity type. For example, if a source system is enabled for Trading Community Members, Items, and Assets, then the source system can be selected as a data source in the Trading Community Members, Items, and Asset import UIs; 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 trading community record.

FAQs for Manage Customers: Define Source Systems

What happens if I allow multiple source system references?

Allowing multiple source system references means that when you import data from a source system you can merge multiple, or duplicate, source system records and create one record in the Oracle Fusion Applications database.

If you do not allow multiple source system references then an Oracle Fusion Applications database record will be created for every source system record. This means that you could potentially create duplicate records in the Oracle Fusion Applications database.

Manage Customers: Import Person and Organization

Trading Community Model Data Import Objects: Explained

Import objects are business entities that can be imported into the trading community model registry, for example, competitors, partners or resource teams. When you create a data import batch you should choose which business entity, or object, you are importing from the batch into the trading community model registry. For example, if you are responsible for resource management, you might want to import objects such as employee resource and resource team.

The import process flow will change according to which object you have selected. There are two import process flows for the following sets of objects:

1. Customer, reference, competitor, and custom party.
2. Employee resource, resource team, partner, and partner contact.

Customer, Reference, Competitor, and Custom Party

When you select these objects you will receive the option to check for duplicates within the import batch before the import, and the option to check for duplicates between the import batch and the trading community model registry before import. You will also be able to choose to preview data before it is imported, specify if addresses will be cleansed before import, and set how many errors you will allow before the import is terminated.

Employee Resource, Resource Team, Partner, and Partner Contact

If you choose to import these objects you will not be able to deduplicate the batch or registry data. However, you will be able to choose to preview data before it is imported, specify if addresses will be cleansed before import, and set how many errors you will allow before the import is terminated.

Defining the Import Process for Customers and Consumers: Points to Consider

Oracle Fusion Applications provides two utilities to import data, File-Based Import and Data Import. File-based Import allows you to import data directly from a CSV or XML file. Data Import imports data from interface tables once the tables are populated by an external process. Data Import cannot be used in a Cloud deployment, because you cannot populate interface tables using an external process in a Cloud deployment. In Cloud deployments, you can only use File-based import to import data. However, even if you are using File-based Import, you can use the Data Import dashboard to monitor the import job and view reports and statistics. You can choose to pause a File-Import batch, and use Data Import work area to configure data quality services, monitor import jobs, view errors, and view batch statistics. Import activities are paused if the **HZ_IMP_PAUSE_FILE_IMPORT** profile option is set to Yes in the Manage Import Profile Options page. If the profile option is set to Yes, all submitted import activities are sent for administrator review and appear in the Data Import Batches Overview page.

Before importing data into the registry, you need to decide if you want to use the data quality services and if so, how you want to configure the data quality services.

The data quality services are:

- Batch deduplication
- Registry deduplication
- Import to registry options

Batch Deduplication

Within the batch deduplication page you can decide if you want to identify and resolve duplicates within the batch that you are importing from the interface

tables. If you want to check for duplicates you need to choose what match configuration rule you want to use to identify duplicates for each entity. Then you need to specify what action will be taken on the persons, organizations, and address duplicates found within the batch. Your specified actions will be performed on the batch before the data is imported into the registry.

Registry Deduplication

Similar to batch deduplication, registry deduplication identifies duplicates between the data in the batch and the data in the registry before the data is imported into the registry. If you want to check for duplicates you need to choose what match configuration rule you want to use to identify duplicates for each entity. Then you need to specify what action will be taken on the persons, organizations, and address duplicates found in the registry deduplication check. Your specified actions will be performed when you import the batch into the registry.

Import to Registry Options

When defining an import process you can decide whether to run the import process in preview mode, or you can choose to load the data directly into the registry without previewing the data. You can also choose to cleanse addresses prior to import, validate addresses in an import batch against geography data, and define an error limit for the batch.

Import Process Mode

You can choose to run the import batch in preview mode, or you can skip the preview and load the data directly into the registry.

If you select to run the batch in preview mode you will be able to review information about the level of duplicates or incorrect addresses in the batch data before the data is actually imported. You will also be able to preview how many records will be created and how many records will be updated for each entity. You can then continue to import the batch, or you can amend the match configuration rules and actions to be taken on the identified duplicates and then rerun the batch to review the data again.

If you do not want to review the batch data before it is imported into the registry, then you can choose to skip the preview and allow the data to be loaded into the registry as soon as preprocessing is complete. You may prefer not to preview the batch data if the data source is frequently used.

Note

The Define Import: Import to Registry page is the only place that you can specify if you want to run the batch in preview mode. Once the option to skip the preview mode is selected, and you submit the batch for processing, you will not be able to review the batch data before it is imported.

Cleanse Addresses

You can choose to validate the addresses in the interface tables before importing them into the registry. The addresses are validated using an integrated third party service that verifies addresses and corrects them if they are incorrect.

Geography Validation

You can choose to validate the addresses in an import batch against geography data before importing them into registry. The addresses are validated against the master reference geography data, according to the geography-based address validation settings for each country. The addresses with validation errors are not imported. However, Geography Name Referencing is run on all imported addresses regardless of this profile option setting.

Error Handling Limit

You can define how many process errors can be generated by the import batch process before the process terminates automatically. Error reports are generated by the application for you to review.

Defining the Import Process for Customers and Consumers: Worked Example

This example demonstrates how to create an import batch, and how to define the import process data quality services you want to use on the data prior to completing the import.

Note

Once an import is completed, the data is loaded into the Trading Community Model registry.

Create an import batch containing customer and consumer objects, load the import data into the interface tables, and configure the data import process so that you can view the batch in preview mode to check that all duplicate data are removed.

Creating an import batch

1. On the Data Import Batches Overview page, click on the **Create Data Import Batch** task located in the task pane.
2. On the Create Data Import Batch page, complete the fields as shown in this table:

Field	Value
Batch Name	Customer Import Batch
Source System	Comma separated values
Object	Customer and Consumer
Estimated Number of Records	300
Batch Description	Import of customer data

3. Click **Save and Close**

4. After creating the import batch, load your data into the interface tables using a standard ETL tool.

Defining the Data Import Process: Batch Deduplication

You want to check the batch for address, organization, and person duplicates; you also want to remove all of these duplicates from the batch.

1. On the Data Import Batches Overview page, highlight the Batch Name, but do not click on the Batch ID URL.
2. Click **Actions** and then click **Import**.
3. On the Define Import: Batch Deduplication page, select the **Check for duplicates within the batch before import**.
4. Complete the fields in the **Select match configuration to determine duplicates within the import batch** section, as shown in this table:

Field	Value
Addresses	Batch Location Basic Duplicate Identification
Organizations	Batch Organization Basic Duplicate Identification
Persons	Batch Person Basic Duplicate Identification

5. Complete the fields in the **Override Default Actions** section, as shown in this table:

Field	Value
Select Action for Persons and Organizations	Remove all duplicates
Select Action for Addresses	Remove all duplicates

6. Click **Next**.

Defining the Data Import Process: Registry Deduplication

You want to check for address, organization, and person duplicates between the import batch and the Trading Community Model registry; you also do not want to import duplicate records.

1. On the Define Import: Registry Deduplication page, select the **Check for duplicates between the import batch and the registry before import**.
2. Complete the fields in the **Select match configuration to determine duplicates within the import batch** section, as shown in this table:

Field	Value
Addresses	Batch Location Basic Duplicate Identification

Organizations	Batch Organization Basic Duplicate Identification
Persons	Batch Person Basic Duplicate Identification

- Complete the fields in the Override Default Actions section, as shown in this table:

Field	Value
Select Action for Persons and Organizations	Do not import duplicate records
Select Action for Addresses	Do not import duplicate records

- Click **Next**.

Defining the Data Import Process: Import to Registry

You want to configure the data import process so that you can view the batch in preview mode, enabling you to review the data after preprocessing. You want to cleanse addresses before they are imported, and specify an error limit for the batch. You also want to validate all addresses in the import batch against geography data. This validates all incoming addresses as per the geography structure and validation level setup in Manage Geographies task. The addresses that are reported as Error during validation are not imported.

- On the Define Import: Import to Registry page, select **Run the batch in preview mode**.
- Select **Cleanse addresses before import**.
- Select **Validate address against geography**.
- In the **Error Limit** field, enter 200.
- Click **Submit**.

Performing What-If Analysis on Data Import Batches: Worked Example

This example demonstrates how to perform What-If analysis on a data import batch that has been processed and has completed with a status of pre-import completed. The match configuration is redefined and the import process is resubmitted. The batch deduplication actions are then amended, and the batch import is completed.

The following table summarizes key decisions for this scenario.

Decisions to Consider	In This Example
Do you want to redefine batch deduplication match configuration?	Yes, a different match configuration is selected for the organizations entity.

Do you want to redefine registry deduplication match configuration?	Yes, a different match configuration is selected for the persons entity.
What actions do you want to take on Persons, Organizations, and Addresses duplicates?	<ul style="list-style-type: none"> • Within registry deduplication, choose Do not import duplicate records for Persons and Organizations. • Within registry deduplication, choose Import duplicate records for Addresses.

Prerequisites

1. The data import batch has been created.
2. The data is uploaded into the interface tables.
3. The batch is imported and has completed with a status of pre-import completed.

Viewing the What-If Analysis

1. On the Data Import Batches Overview page, click on the batch ID URL.
2. On the Edit Data Import Batch page, review the summary and import process performance information. Click **Import Details** to open the What-If analysis page.
3. On the Import Process Details page, click the **Batch Deduplication** tab. Check that you are satisfied with the batch deduplication results.
4. On the Import Process Details page, click the **Registry Deduplication** tab. Check that you are satisfied with the registry deduplication results.
5. On the Import Process Details page, click the **Address Cleansing** tab. Check that you are satisfied with the address cleansing results.

Redefining the Match Configuration and Resubmitting the Import Process

The results of the batch and registry deduplication are not as expected and so the match configurations need to be redefined.

1. On the Import Process Details page, click **Cancel**.
2. On the Data Import Batches Overview page, click on the batch name. Click **Actions** and then click **Import**.
3. On the Define Import: Batch Deduplication page, choose a different match configuration for the organizations entity. Click **Next**.
4. On the Define Import: Registry Deduplication page, choose a different match configuration for the persons entity. Click **Next**.
5. On the Define Import: Import Registry page, click **Submit**

Changing the Action for Duplicates within the What-If Analysis

You want to view the What-If analysis for the new match configurations that you selected for the batch.

1. On the Data Import Batches Overview page, click on the batch ID URL.

2. On the Edit Data Import Batch page, review the summary and import process performance information. Click **Import Details** to open the What-If analysis page.
3. The new match configurations have produced satisfactory results, but you would like to change the actions that will be carried out on the duplicates. On the Import Process Details page, click the **Registry Deduplication** tab.
4. For the Persons and Organizations duplicates, choose **Do not import duplicate records** from the choice list.
5. For the possible duplicates for Addresses, choose **Import duplicate records** from the choice list.
6. Click **Complete Import**.

FAQs for Manage Customers: Import Person and Organization

Can I redefine the data import process for an already imported batch and reimport it?

Yes. If the data is still available in the interface tables and the batch status is **Preimport Completed**, **Completed with Errors**, **Error**, or **Terminated when Error Limit Reached**, then you can redefine the data import process and reimport the batch. However, once a batch has been successfully imported then you will not be able to reimport the batch, even if the data is present in the interface tables.

How can I view the errors that occurred during preimport processing?

You can view any errors that occurred after submitting the batch for import by selecting the batch in the data import batches Overview page, and then click **Report**.

What happens if I purge a data import batch?

You permanently remove all records in the batch from the import interface tables. You should purge batches after the batch has been imported successfully and you are sure that the data in the interface tables is no longer required.

Purging the interface tables improves import performance. To archive imported data, you should copy the data to a set of custom tables.

Why can't I purge a data import batch?

Data import batches cannot be purged when they have a status of **Processing**. Purging a batch purges the batch data in the interface tables and so cannot be carried out while the batch is importing the data from the interface tables into the registry.

Why did I receive a message that the data enrichment process was declined?

The Oracle Fusion Trading Community Data Quality service may be unavailable, or you may not have the necessary licenses for the Data Quality service.

Can I try and import a batch again?

Yes. You can reimport a batch to correct validation errors if the import process status is either **Preimport Completed**, **Completed With Errors**, **Error**, or **Terminated When Error Limit Reached**.

Why can't I view the import process details?

The import process details are only available when you have selected to run the batch in preview mode, and you have submitted the batch for import.

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 (Oracle Fusion Applications Edition) and the Oracle Fusion Applications Extensibility Guide.

Report Permissions

- You can restrict access to specific reports for specific users, and this security is not limited to the Reports and Analytics pane. Refer to the

Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition).

See: Assigning Permissions

Oracle Business Intelligence Publisher Reports Submission

- Oracle BI Publisher reports must be registered as processes with Oracle Enterprise Scheduler to be enabled for scheduling. This registration also enables a Schedule link for the report in the Reports and Analytics Pane. Refer to the Oracle Fusion Applications Extensibility Guide, and perform the following steps in the specified order.
 - Create an Oracle Enterprise Scheduler job definition for the report.
See: Tasks Required to Run Custom Reports with Oracle Enterprise Scheduler Service
 - Specify the job definition details in the report's properties.
See: Enabling Reports for Scheduling from the Reports and Analytics Pane

FAQs for Map Reports to Work Areas

How can I set up the Reports and Analytics pane for all users?

You can remove any currently mapped report from the Reports and Analytics pane, or add mappings to reports from the Oracle Business Intelligence (BI) Presentation catalog. To access the setup, click **Edit Settings** in the Reports and Analytics pane, or use the Map Reports to Work Areas task in the Setup and Maintenance work area. If you do the former, then you can set up only the Reports and Analytics pane on the work area that you are in.

If you do the latter, then you can select a work area to set up. If you do not see the desired work area, most likely you do not have access to it due to security. You can request to be granted a role that has access to the work area, or another administrator or business user with access to the work area can be granted the Reports and Analytics Region Administration Duty to be able to map reports to the work area.

Tip

On the Map Reports to Work Areas page only, you can also use the Synchronize button to remove mappings to reports that are no longer in the catalog, for all work areas at once.

Any changes you make in either UI apply to all users with access to the mapped work area.

Why can't I see reports when mapping reports to work areas for the Reports and Analytics pane?

It is possible that there are no reports currently mapped to the work area that you select in the Map Reports to Work Areas page. Alternatively, reports are mapped, but you do not see them due to security.

Similarly, in the list of all available reports from the catalog, you can see only the reports that you have access to. You can request to be granted a role that has access to the reports that you want to map, or another administrator or business user with access to those reports can be granted the Reports and Analytics Region Administration Duty to be able to map reports to work areas.

Why can't I see reports when I edit settings for the Reports and Analytics pane?

In the Edit Settings window, you may not be able to see a currently mapped report because you do not have access to it due to security.

Similarly, in the list of all available reports from the catalog, you can see only the reports that you have access to. You can request to be granted a role that has access to the reports that you want to map, or another administrator or business user with access to those reports can be granted the Reports and Analytics Region Administration Duty to be able to map reports to work areas.

Set Watchlist Options

Watchlist Setup: Points to Consider

For all users across the site, you can disable or enable predefined Watchlist categories and items, edit their names, and determine how often item counts refresh. You cannot delete predefined Watchlist categories and items, nor create any for the site. Users can create their own Watchlist items through saved searches.

Access the Set Watchlist Options page by starting in the Setup and Maintenance Overview page and searching for the Set Watchlist Options task.

Disabling Predefined Categories and Items

Use the Set Watchlist Options page to enable or disable predefined Watchlist categories and items. Disabling any category or item also disables associated processes involved in calculating the Watchlist item counts for all users. These processes include creating data caches, performing security checks, invoking services across domains, running queries, and so on.

An item with the **Predefined** type represents the actual predefined Watchlist item that appears in the Watchlist. If you disable this type of Watchlist item, then:

- The item is not available for users to display in their watchlist
- The item is removed from any watchlist where it is currently displayed

A Watchlist item with the **User-created saved search** type does not appear in the Watchlist; it controls the display of the **Manage Watchlist** button or menu item in pages with saved searches. If you disable this type of Watchlist item, then:

- The **Manage Watchlist** option is not available to users in the corresponding work area, so users cannot use their own saved searches as

Watchlist items. A message is displayed to users when they try to use this option.

- Any user-defined saved searches from that work area already used as Watchlist items are no longer available in the users' watchlist. The user-defined saved searches are still available to be used for searching, but not for the Watchlist.

If you disable a Watchlist category, then the category is not available for users to include in their watchlist, and all Watchlist items within the category are also disabled.

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.

Define Order Promising and Perform Data Collections

Manage Planning Source Systems

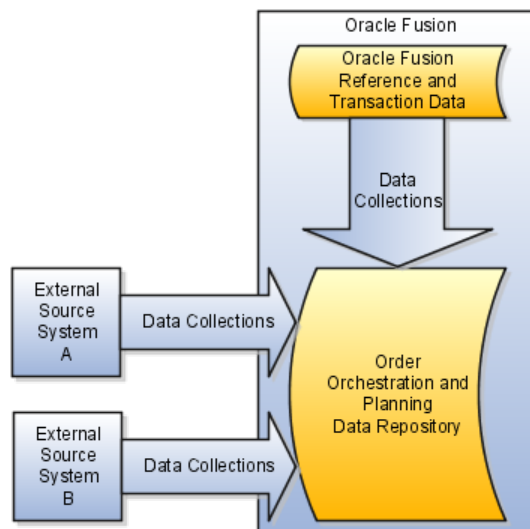
Managing Data Collection Source Systems: Explained

To populate the order orchestration and planning data repository, you collect data from external source systems, such as external fulfillment source systems and external order capture source systems, and from the Oracle Fusion source system. You manage which source systems are data collection source systems by defining collections parameters and enabling which source systems allow collections.

You manage two categories of source systems for data collections:

- External source systems
- The Oracle Fusion source system

The following figure illustrates data collections from three source systems. Two of the source systems are external source systems. One of the source systems is the Oracle Fusion source system.



External Source Systems

Your business may have many external fulfillment and external order capture source systems. For each external source system from which you need to collect data to include in the order orchestration and planning data repository, define the data collection parameters, and enable the source system for collections. For the Version data collection parameter, the choices are Other or Oracle Fusion.

The Oracle Fusion Source System

The order orchestration and order promising processes use data stored in the order orchestration and planning data repository. Some of the data that needs to be in the repository originates in the Oracle Fusion source system. To collect data from the Oracle Fusion source system, include the Oracle Fusion source system as a source system for data collection. Define the data collection parameters for the Oracle Fusion source system, and enable the source system for collections.

Defining Data Collection Parameters: Points to Consider

For each system from which you intend to collect data to populate the order orchestration and planning data repository, you define and maintain the source system data collection parameters.

For each source system, you complete the following for the data collection parameters:

- Specify the time zone.
- Specify the version, order orchestration type, and planning type.
- Define the number of database connections, parallel workers, rows per processing batch, and cached data entries.
- Enable collections allowed.
- Enable data cross-referencing.

Specify the Time Zone

You must specify the time zone for the source system because the time stamps contained in collected data are converted from the time zone used in the source system to the time zone used for all data stored in the order orchestration and planning data repository. Using the same time zone for all data stored in the order orchestration and planning data repository facilitates correct results when calculations are performed using attributes that store dates. For example, if the source system uses the US Eastern time zone, but the order orchestration and planning data repository stores all data in the US Pacific time zone, then a supply with a due date and time of July 10th 04:00 PM in the source system is stored in the order orchestration and planning data repository with a due date of July 10th 01:00 PM.

Specify the Version, Order Orchestration Type, and Planning Type

You must define one, and only one, source system with the Version attribute equal to Oracle Fusion and the Order Orchestration Type attribute equal to Order Orchestration.

You may define many source systems with the Version attribute equal to Other. For the source systems with the Version attribute equal to Other, the Order Orchestration Type attribute can equal Fulfillment or Order Capture and the Planning Type attribute can equal Fulfillment. Any combination of these values is allowed to describe the purpose of the source system, but you must provide a value for at least one of these type parameters. These parameters do not impact the behavior of the collections process.

Note

Once you have saved a system with the Version attribute equal to Oracle Fusion, you cannot change the value for the Version attribute.

Note

You cannot change the version of a source system from Others to Fusion. You must delete the planning source system definition by scheduling the Delete Source Configuration and All Related Data process. The Delete Source Configuration and All Related Data process performs multiple steps. First the process deletes all data previously collected from the source system. After deleting the collected data, the process deletes the planning source system definition and collection parameters. After the Delete Source Configuration and All Related Data process completes, you must redefine the planning source system definition on the Manage Planning Source Systems page.

Define the Number of Database Connections, Parallel Workers, Rows Per Processing Batch, and Cached Data Entries

These parameters affect the usage of system resources. The table below defines what each parameter does and provides guidelines for setting it.

Parameter	What the Parameter Does	A Typical Value for the Parameter
Number of Database Connections	Defines the maximum number of database connections the source server can create during the collection process. This controls the throughput of data being extracted into the Source Java program.	10
Number of Parallel Workers	Defines the maximum number of parallel workers (Java threads) used to process the extracted data. The number here directly impacts the amount of CPU and memory used during a collection cycle.	30
Number of Rows per Processing Batch	Define the number of records to process at a time. The idea is to allow the framework to process data in byte-size chunks. A batch too small may cause extra overhead while a batch too big might peak out memory or network bandwidth.	10,000

Cached Data Entries in Thousands	During data collections, various lookup and auxiliary data are cached in the collection server to support validation. For example, currency rate may be cached in memory. This parameter controls the maximum number of lookup entries cached per lookup to prevent the server from occupying too much memory.	10,000
---	--	--------

Enable Collections Allowed

Before enabling a source system for collections, ensure your definition of the other parameters are complete for the source system. Ensure you have defined values for all applicable attributes, and where applicable, you have enabled organizations for collections or for ATP Web services.

Enable Data Cross-Referencing

When you enable a source system for data cross-reference, the data collections from the source system perform additional processing steps to check for and to cross-reference data during data collections. You must enable cross-referencing for Order Capture source systems.

Enabling Organizations for Data Collections: Points to Consider

From the list of organizations for each source systems, you designate which organizations will have their data collected when a collections process collects data from the source system.

Deciding Which Organizations to Enable for Collections

To determine which organizations to enable for collections, analyze the sourcing strategies for your company, the type of organization for each organization in the list, and any other business requirements that would determine whether system resources should be expended to collect data from that organization. If the data from that organization would never be used by order promising or order orchestration, no need to collect the data.

For example, consider a scenario where the list of organizations for a source system includes 20 manufacturing plants and 10 distribution centers. Because the business requirements specify that the movements of materials from the manufacturing plants to the distribution centers are to be controlled separately from order orchestration and order promising, there are no sourcing rules that include transferring from one of the manufacturing plants. For this scenario, you would only enable the 10 distribution centers for collections.

Enabling Organizations for ATP Web Service: Points to Consider

You enable the available-to-promise (ATP) Web Service to enable Oracle Fusion Global Order Promising to invoke an external order promising engine to determine a date and quantity available for fulfillment lines to be shipped from a specific organization.

Deciding Which Organizations to Enable for ATP Web Service

Your business requirements may require you to obtain the available-to-promise dates and available-to-promise quantities from external fulfillment systems for fulfillment lines to be shipped from specific organizations. To implement such a requirement, you enable the ATP Web Service for each organization subject to the requirement.

When a fulfillment line is received with a ship-from organization that is equal to an organization for which ATP Web Service has been enabled, Oracle Fusion Global Order Promising invokes the order promising engine of the applicable external system to determine an available-to-promise date and an available-to-promise quantity. For example, if the ATP Web Service has been enabled for your Lima organization, when fulfillment lines are received with Lima specified for the ship-from organization, an external order promising engine is invoked to provide an available-to-promise date and an available-to-promise quantity for these fulfillment lines.

There are some rare cases where an organization would be enabled for both collections and for ATP Web Service. For example, consider the scenario where there are different sourcing strategies for two items, Item X and Item Y. Item X is only purchased from the Chicago organization, but the Chicago organization can transfer Item X from the Seattle organization, and the sourcing rules have specified that the order promising process should check for inventory at the Seattle organization for Item X when Chicago is out of stock. The Seattle organization must be enabled for collections because the data from the Seattle organization must be included in the Order Orchestration and Planning data repository for use by the order promising process when it needs to check for available supply of Item X at the Seattle organization to transfer to the Chicago organization. Item Y is purchased directly from Seattle, must be made to order, and the order promising process must invoke an external ATP Web service to determine when the Item Y could be made. In this case, you would also enable the Seattle organization for ATP Web service.

FAQs for Planning Source Systems

Can I add a new source system to the list of data collection source systems?

No. You cannot add additional source systems when managing source systems for data collections for the order orchestration and planning data repository.

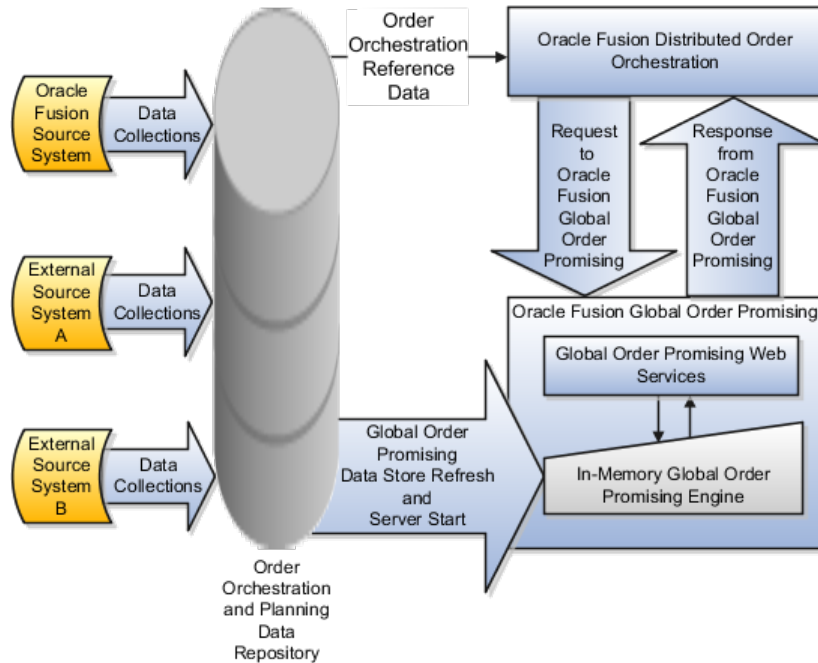
Source systems must first be defined in the Trading Community Model. For the system to be listed as one of the systems from which to choose from when managing source systems, the definition of the system in the Trading Community Model must enable the system for order orchestration and planning.

Data Collections, Order Orchestration, and Order Promising: How They Fit Together

You perform data collections to populate the order orchestration and planning data repository. The collected data is used by Oracle Fusion Distributed Order Orchestration and Oracle Fusion Global Order Promising.

The following figure illustrates that the order orchestration and planning data repository is populated with data from external source systems and from the

Oracle Fusion source system when you perform data collections. Oracle Fusion Distributed Order Orchestration uses some reference data directly from the repository, but the Global Order Promising engine uses an in-memory copy of the data. After data collections are performed, you refresh the Global Order Promising data store with the most current data from the data repository and start the Global Order Promising server to load the data into main memory for the Global Order Promising engine to use. When Oracle Fusion Distributed Order Orchestration sends a scheduling request or a check availability request to Oracle Fusion Global Order Promising, the Global Order Promising engine uses the data stored in main memory to determine the response.



Data Collections

You perform data collections to populate the order orchestration and planning data repository with data from external source systems and from the Oracle Fusion source system.

Order Orchestration

Oracle Fusion Distributed Order Orchestration uses some reference data directly from the order orchestration and planning data repository. You must perform data collections for the order orchestration reference entities even if you are not using Oracle Fusion Global Order Promising.

Important

Before collecting data from an Oracle Fusion source system, you must define at least one organization for the source system. After you have defined at least one organization for the source system, you must update the organization list for the source system on the Manage Planning Source Systems page or Manage Orchestration Source Systems page, and enable at least one organization for collections. If there are no organizations enabled for collections when a collections process runs, the collections process will end with an error.

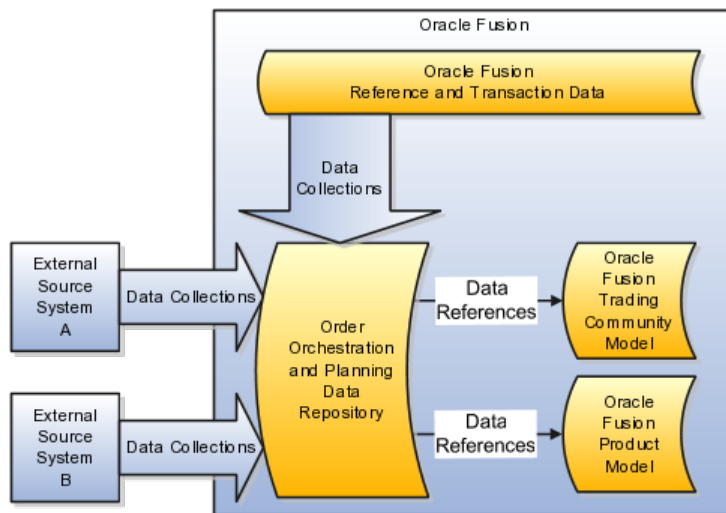
Order Promising

The Global Order Promising engine uses an in-memory copy of the data from the order orchestration and planning data repository. When Oracle Fusion Distributed Order Orchestration sends a scheduling request or a check availability request to Oracle Fusion Global Order Promising, the Global Order Promising engine uses the data stored in main memory to determine the response to send back to order orchestration. After a cycle of data collections is performed, you refresh the Global Order Promising data store with the most current data from the data repository and start the Global Order Promising server to load the data into main memory for the Global Order Promising engine to use.

Collecting Data for the Order Orchestration and Planning Data Repository: Explained

The order orchestration and planning data repository provides a unified view of the data needed for order orchestration and order promising. You manage data collection processes to populate the data repository with data collected from external source systems and from the Oracle Fusion source system. You manage the data collection processes to collect the more dynamic, transaction data every few minutes and the more static, reference data on a daily, weekly, or even monthly schedule. The data collected into the data repository contains references to customer data managed in the Oracle Fusion Trading Community Model and to product data managed in the Oracle Fusion Product Model. The data managed in these models is not collected into the order orchestration and planning data repository.

The following figure illustrates that the order orchestration and planning data repository is populated with data collected from external source systems and from the Oracle Fusion source system. The data repository does not contain data managed by the Oracle Fusion Trading Community Model and the Oracle Fusion Product Model. The data collected into the data repository references data managed in these models.



When you plan and implement your data collections, you determine:

- Which entities you collect from which source systems
- The frequency of your collections from each source system
- Which data collection methods you will use to collect which entities from which source systems
- The sequences of your collections

Consider these categories of data when you plan your data collections:

- Data collected for order promising
- Data collected for order orchestration
- Data not collected into the order orchestration and planning data repository

Data Collected for Order Promising

The following categories of data are collected and stored to support order promising:

- Existing supply including on-hand, purchase orders, and work orders
- Capacity including supplier capacity and resource capacity
- Related demands including work order demands and work order resource requirements
- Planned supply including planned buy and make orders
- Reference data including calendars, transit times, and routings

For a specific collection, the data that is collected is determined by the data collection entities specified for that collection.

Important

After performing data collections, you must refresh the Order Promising data store and start the Order Promising engine to ensure the engine is using the data most recently collected.

Data Collected for Order Orchestration

The following reference data is collected and stored to support order orchestration:

- Warehouses (Organization Parameters)
- Units of Measure
- Carriers
- Currencies
- Currency Conversions
- Shipping Methods
- Order Orchestration Reference Objects
 - Payment Term
 - Accounting Rule
 - Invoicing Rule
 - Sales Credit Type

- AR Receipt Method
- Document Category
- Service Level
- Mode of Transport
- Tax Classification Code
- Return Reason
- Shipment Priority
- Payment Method
- Freight Terms
- Tax Exemption Reasons
- Activity Types
- FOB

Tip

Use the Review Planning Collected Data page or the Review Order Orchestration Collected Data page to explore many of the entities and attributes collected for the order orchestration and planning data repository.

Data Not Collected into the Order Orchestration and Planning Data Repository

Data collected into the order orchestration and planning data repository includes attributes, such as customer codes, that refer to data not collected into the data repository. Most of the data references are to data in the Oracle Fusion Trading Community Model or in the Oracle Fusion Product Model. Some of the data references are to data outside the models, such as item organizations and inventory organizations. To manage data collections effectively, especially the sequences of your collections, you must consider the data dependencies created by references to data not collected into the data repository.

References to data in the Oracle Fusion Trading Community Model include references to the following:

- Source systems
- Geographies and zones
- Customers
- Customer sites

References to data in the Oracle Fusion Product Model include references to the following:

- Items, item relationships, and item categories
- Item organization assignments
- Structures

Data Collection Entities: Explained

When you collect data for the order orchestration and planning data repository, you specify which of the data collection entities to collect data for during each collection. When you plan your data collections, you plan which entities to collect from which source systems and how frequently to collect which entities. One of the factors you include in your planning considerations is the categorizations of each entity. One way entities are categorized is as reference entities or transaction entities. You typically collect transaction entities much more frequently than reference entities.

Another way entities are categorized is as source-specific entities or global entities. For global entities the order in which you collect from your source systems must be planned because the values collected from the last source system are the values that are stored in the data repository.

When you plan your data collections, you consider the following categorizations:

- Source-specific entities
- Global entities
- Reference entities
- Transaction entities

You also consider which entities can be collected from which types of source systems using which data collection methods as follows:

- Entities you can collect from the Oracle Fusion source system and from external source systems
- Entities you can collect only from external source systems

Source-Specific Entities

When you collect data for a source-specific entity, every record from every source system is stored in the order orchestration and planning data repository. The source system association is maintained during collections. The data stored in the data repository includes the source system from which the data was collected.

For example, you collect suppliers from source system A and source system B. Both source systems contain a record for the supplier named Hometown Supplies. Two different supplier records will be stored in the data repository for the supplier named Hometown Supplies. One record will be the Hometown Supplies supplier record associated with source system A and the second record will be the Hometown Supplies supplier record associated with source system B.

The majority of the data collections entities are source-specific entities.

Global Entities

When you collect data for a global entity, only one record for each instance of the global entity is stored in the order orchestration and planning data repository. Unlike source-specific entities, the source system association is not maintained during collections for global entities. The data stored in the data repository for global entities does not include the source system from which the data was collected. If the same instance of a global entity is collected from more than one source system, the data repository stores the values from the last collection.

For example, you collect units of measure (UOM) from three source systems and the following occurs:

1. During the collection of UOM from source system A, the Kilogram UOM is collected.

This is first time the Kilogram UOM is collected. The Kilogram record is created in the data repository.

2. During the collection of UOMs from source system B, there is no collected UOM with the value = Kilogram

Since there was no record for the Kilogram UOM in source system B, the Kilogram record is not changed.

3. During the collection of UOMs from source system C, the Kilogram UOM is also collected.

Since the collections from source system C include the Kilogram UOM, the Kilogram record in the data repository is updated to match the values from source system C.

The following entities are the global entities:

- Order orchestration reference objects
- Units of measure (UOM) and UOM conversions
- Demand classes
- Currency and currency conversion classes
- Shipping methods

Tip

When you collect data for global entities from multiple source systems, you must consider that the last record collected for each occurrence of a global entity is the record stored in the order orchestration and planning data repository. Plan which source system you want to be the source system to determine the value for each global entity. The source system that you want to be the one to determine the value must be the source system that you collect from last.

Reference Entities

Reference entities are entities that define codes and valid values that are then used regularly by other entities. Units of measure and demand classes are two examples of reference entities. Reference entities are typically static entities with infrequent changes or additions. Whether an entity is reference entity or a transaction entity does not impact how it is stored in the order orchestration and planning data repository.

You consider whether an entity is a reference entity or a transaction entity when determining which collection method to use to collect data for the entity. You typically use the staging tables upload method to collect data for reference entities from external source systems. You typically used the targeted collection method to collect data for reference entities from the Oracle Fusion source system unless the reference entity is one of the entities for which the targeted collection method is not possible.

Transaction Entities

Transaction entities are the entities in the data repository that store demand and supply data. Because the data for transaction entities changes frequently, you typically use the web services upload method to collect data for transaction entities from external source systems. You typically use the continuous collection method to collect data for transaction entities from the Oracle Fusion source system.

Entities You Can Collect From the Oracle Fusion Source System and From External Source Systems

Many of the data collection entities can be collected from both types of sources systems. For the following entities you can use any of the collections methods:

- Approved supplier lists
- Calendars
- Calendar associations
- Interlocation shipping networks
- Item costs
- On hand
- Organization parameters
- Purchase orders and requisitions
- Subinventories
- Suppliers
- Units of measure

For the following entities you can only use the Web service upload method to collect data from external source systems:

- Currencies
- Order orchestration reference objects
- Shipping methods

Entities You Can Collect only From External Source Systems

Many of the data collection entities can be only collected from external sources systems. For these entities, you can use both methods for collecting data from external source systems. Remember to consider frequency of change and volume of data in your considerations of which methods to use to collect which entities. The following are the entities you can only collect from external sources systems:

- Customer item relationships
- Demand classes
- Planned order supplies
- Routings
- Resources
- Resource availability
- Sourcing
- Supplier capacities
- Work-in-process supplies

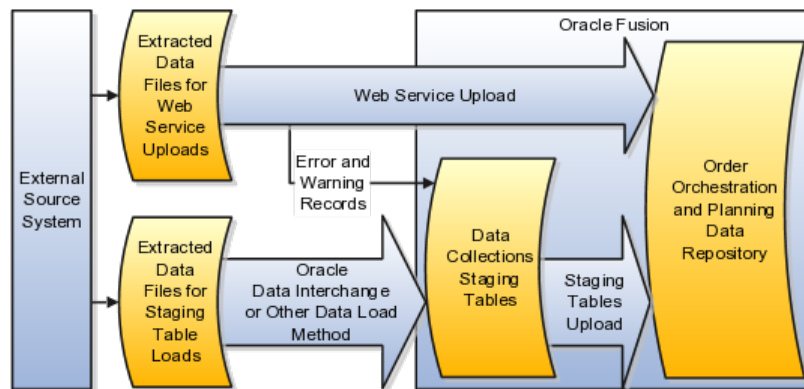
- Work-in-process component demands
- Work-in-process resource requirements

Collect Order Promising Reference and Transaction Data

Data Collection Methods for External Source Systems: Explained

To populate the order orchestration and planning data repository with data collected from external source systems, you use a combination of two data collection methods. The two methods are Web service uploads and staging tables uploads.

The following figure illustrates the two data collection methods, Web service uploads and staging tables uploads, used to collect data from external source systems. The figure illustrates that both methods require programs to be written to extract data from the external source systems. For Web service uploads, you load the data from the extracted data files directly into the order orchestration and planning data repository. Any records with errors or warnings are written to the data collections staging tables. For staging table uploads, you load the data from the extracted data files into the data collections staging tables, and then you use the Staging Tables Upload program to load the data from the staging tables into the data repository.



You determine which entities you collect from which source systems and at what frequency you need to collect the data for each entity. The data for different entities can be collected at different frequencies. For example, supplies and demands change frequently, so collect data for them frequently. Routings and resources, are more static, so collect data for them less frequently.

Which data collection method you use for which entity depends upon the frequency of data changes as follows:

- Web service upload
 - Use for entities with frequent data changes.
- Staging tables upload
 - Use for entities with more static data.

Web Service Upload Method

Use the Web service upload method for entities that change frequently, such as supply and demand entities. You determine the frequency of collections for each

entity. For certain entities, you may implement Web services to run every few minutes. For other entities, you may implement Web services to run hourly.

To implement and manage your Web service uploads, you must design and develop the processes and procedures to extract the data in the format needed by the data collection web services. For more information regarding the data collection Web services, refer to the Oracle Enterprise Repository. For additional technical details, see Oracle Fusion Order Promising Data Collection Staging Tables and Web Service Reference, document ID 1362065.1, on My Oracle Support at <https://support.oracle.com>.

Staging Tables Upload Method

Use the staging tables upload method for entities that do not change frequently, such as routings and resources. You determine the frequency of collections for each entity. You may establish staging table upload procedures to run daily for some entities, weekly for some entities, and monthly for other entities.

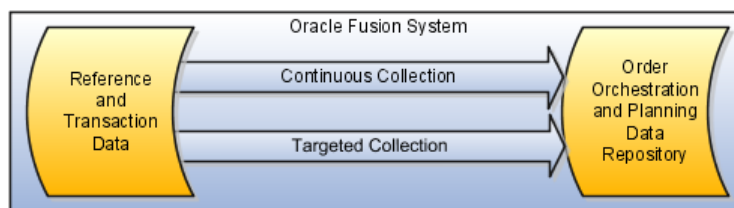
To implement and manage your staging table uploads, you must develop the processes and procedures you use to extract data from an external source system. You use Oracle Data Interchange, or another data load method, to load the extracted data into the data collection staging tables. For additional technical details, such as the table and column descriptions for the data collection staging tables, see Oracle Fusion Order Promising Data Collection Staging Tables and Web Service Reference, document ID 1362065.1, on My Oracle Support at <https://support.oracle.com>.

For the final step of the staging tables upload method, you initiate the Load Data from Staging Tables process from the Manage Data Collection Processes page or via the Enterprise Scheduling Service.

Data Collection Methods for the Oracle Fusion Source System: Explained

To populate the order orchestration and planning data repository with data collected from the Oracle Fusion source system, you use a combination of two data collection methods: continuous collection and targeted collection. You typically use continuous collection for entities that change frequently and targeted collection for entities that are more static.

The following figure illustrates the two data collection methods, continuous collection and targeted collection, used in combination to collect data from the Oracle Fusion source system.



Continuous Collection

When you use the continuous collection method, you are only collecting incremental changes, and only for the entities you have included for continuous

collection. Because continuous collection only collects incremental changes, you usually set up the continuous collection to run frequently, such as every five minutes.

Note

Prior to including an entity for continuous collection, you must have run at least one targeted collection for that entity.

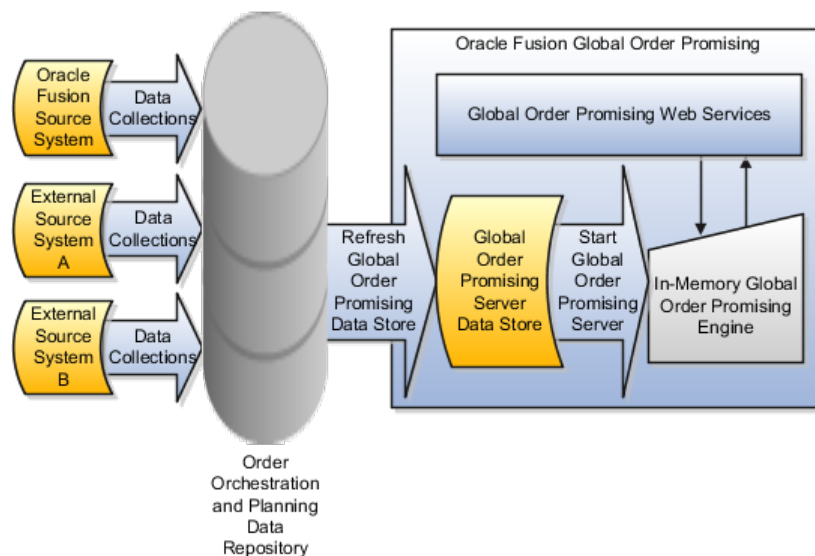
Targeted Collection

When you collect data using the targeted collection method, you specify which entities to include in the targeted collection. For the included entities, the data in the data repository that was previously collected from the Oracle Fusion source system is deleted and replaced with the newly collected data. The data for the entities not included in the targeted collection is unchanged. You typically use the targeted collection method to collect data from entities that do not change frequently.

Refreshing the Global Order Promising Engine: Explained

The Global Order Promising engine is an in-memory engine that uses an in-memory copy of the data collected into the order orchestration and planning data repository. To ensure the in-memory data reflects the latest supply and demand data collected into the data repository, you should refresh the Global Order Promising data store and start the Global Order Promising server at least once a day.

The following figure illustrates that you perform data collections to populate the order orchestration and planning data repository with current data from multiple source systems. After you complete a cycle of data collections, you refresh the Global Order Promising data store with the latest data from the data repository. After you refresh the Global Order Promising data store, you start the Global Order Promising server to load a copy of the refreshed data from the data store into main memory.



To refresh the in-memory copy of the collected data with the most recently collected data, perform these two steps:

1. Refresh the Global Order Promising data store.
2. Start the Global Order Promising server.

Refresh the Global Order Promising Data Store

To refresh the Global Order Promising data store, complete these steps:

1. Navigate to the Schedule New Process page by following this navigation path:
 - a. Navigator
 - b. Tools
 - c. Schedule Processes
 - d. Schedule New Process
 - e. Click the more link
2. Select the **Schedule Processes** link.
3. Click the **Submit New Request** button.
4. In the popup window, select Job for the type.
5. Search for and select the process named RefreshOpDatastore.
6. Select the entities you want to refresh and submit the job.

Start the Global Order Promising Server

To start the Global Order Promising server, you use an Oracle Fusion Global Order Promising instantiation of Oracle Enterprise Manager.

You do not need to stop the server before you start it. If the Global Order Promising server is already running when you start the Global Order Promising server, the Global Order Promising engine currently in memory continues to run until the start process is complete. The Start Global Order Promising Server process updates another engine with the current data from the Global Order Promising Server data store. When the updated engine comes up, the existing engine with the old data is automatically shut down.

Important

The Current Date attribute stored within the Global Order Promising engine is also updated when you start the Global Order Promising server. If the Global Order Promising engine is not updated at least once a day, the Global Order Promising engine may have a wrong current date, and there may be issues with promising results.

Note

You also use an Oracle Fusion Global Order Promising instantiation of Oracle Enterprise Manager to monitor performance of the Global Order Promising server, to access log files, and to stop the server when necessary.

Manage Planning Data Collection Processes

Managing Data Collection Processes: Overview

For your data collections from the Oracle Fusion source system, you use the Manage Planning Data Collection Processes page or the Manage Orchestration Data Collection Processes page. From these pages you perform the following:

- Manage your continuous collections from the Oracle Fusion source system.
- Manage your collections destination server.
- Perform your targeted collections from the Oracle Fusion source system.

For your data collections from external source systems, most of the management of your Web services uploads and staging tables uploads is performed external to the Oracle Fusion application pages. If you choose to perform staging tables uploads, you initiate the Perform Data Load process from the Manage Planning Data Collection Processes page, from the Manage Orchestration Data Collection Processes page, or from the Oracle Fusion Enterprise Scheduler.

Continuous Collection Publish Process: Explained

To enable continuous collections, you must set up the publish data processes for the Oracle Fusion source system. The publish process performs the incremental data collections from the Oracle Fusion source system. You can start, stop, and pause the publish process. To review statistics regarding the publish process, view process statistics from the **Actions** menu on the Continuous Collection - Publish tab on the Manage Planning Data Collection Processes page or the Manage Orchestration Data Collection Processes page.

Note

Because continuous collections only collect net changes, you must perform at least one targeted collection for an entity before you include the entity for continuous collections.

Publish Process Parameters: Points to Consider

You define the publish process parameters to determine the frequency and scope of the continuous collections publish process.

You define the frequency and scope of continuous collections by specifying the following:

- Process Parameters
- Process Entities

Process Parameters

You determine how frequently the continuous collections publish process executes by specifying the frequency in minutes. The continuous collections publish process will publish incremental changes based on the frequency that was defined when the publish process was last started.

You determine which organizations will be included in the set of organizations for which data is collected by specifying an organization collection group. You can leave it blank if you want data collected from all organizations.

Process Entities

You determine which entities are collected during the continuous collections cycles by selecting which entities you want included in the collections. The continuous collections publish process collects incremental changes for the business entities that were included when the publish process was last started.

Collections Destination Server: Explained

The collections destination server is applicable to all four data collection methods. For the continuous collections method the collections server is the subscriber to the continuous collections publish process. From the Actions menu on the Collections Destination Server tab you can access a daily statistic report with statistics regarding each of the collection methods. You also can access a data collections summary report.

Destination Server Collections Parameters: Points to Consider

The collection parameters are initially set to what was defined for the Oracle Fusion system when your planning source systems or order orchestration source systems were initially managed. You can fine tune the parameters for your data collections.

Data Collection Parameters

The data collection parameters affect the usage of system resources. This table define what each parameter does and provides guidelines for setting it.

Parameter	What the Parameter Does	A Typical Value for the Parameter
Number of Database Connections	Defines the maximum number of database connections the source server can create during the collection process. This controls the throughput of data being extracted into the Source Java program.	10
Number of Parallel Workers	Defines the maximum number of parallel workers (Java threads) used to process the extracted data. The number here directly impacts the amount of central processing units and memory used during a collection cycle.	30
Cached Data Entries in Thousands	During data collections, various lookup and auxiliary data are cached in the collection server to support validation. For example, currency rate may be cached in memory. This parameter controls the maximum number of lookup entries cached per lookup to prevent the server from occupying too much memory.	10,000

Cross-Referencing Data During Data Collections: Explained

When you collect data from multiple source systems, you often collect a variety of values for the same instance of an entity. You cross-reference data during data collections to store a single, agreed value in the order orchestration and planning data repository for each instance of a global entity.

The following information explains why you might need to cross-reference your data during data collections, and what you need to do to implement cross-referencing:

- Cross-reference example
- Cross-reference implementation

Cross-Reference Example

The following table provides an example of why you might need to cross-reference your data during data collections. In the example, the Kilogram unit of measure is collected from two source systems. The source systems use a different value to represent kilogram. You decide to store kg for the value for Kilogram in the order orchestration and planning repository.

Source System	Collections Entity	Source Value	Target Value
System A	Unit of measure	kilogram	kg
System B	Unit of measure	k.g.	kg

Cross-Reference Implementation

To implement cross-referencing, you must complete the following actions:

1. Decide which business object to enable for cross-referencing.
2. For each object, work with business analyst to decide which values to map to which other values.
3. Update the domain value maps using one of two methods: the Oracle Fusion SOA Composer or the Oracle Fusion Middleware Domain Value Map user interface. The SOA Composer is typically considered easier to use.
4. On the Manage Planning Data Collection Processes page, enable the corresponding entity for cross-referencing.
5. Determine an ongoing procedure for adding new values into the domain value map when new values occur for a business object.

Notes regarding usage of the Oracle Fusion SOA Composer:

- Depending upon how the SOA suite is installed at your location, you should have the SOA Designer role assigned in order to edit the data value maps.
- Choose DVM in MscCollConfigurationXReferenceComposite.
- You must save your changes and commit your changes.

- The APS Weblogic server must be bounced before the changes are visible to the collections engine.
- For documentation regarding using the SOA Composer, refer to the Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite document, Using Oracle SOA Composer with Domain Value Maps section.

FAQs for Manage Planning Data Collection Processes

Can I use continuous collection to collect item costs?

The continuous collection data collection method is partially supported for item costs. Item costs are collected in the next incremental collection cycle for previously existing items when one or more item organization attributes in addition to item cost have changed.

When a new item is defined, the item cost for the new item is not collected in the next incremental collection cycle. If an existing item is not changed other than an update to the item cost, the item cost change is not picked up in the next incremental collection cycle.

Tip

If items are added frequently, item costs are changed frequently, or both, then targeted collection of item costs should be routinely performed, perhaps once a day.

Perform Planning Data Collections

Loading Data into the Data Collections Staging Tables Using Oracle Data Integrator: Explained

To use the staging tables upload method, you must load the data you extract from your external source systems into the staging tables. You can use Oracle Data Integrator to load the extracted data into the staging tables.

If you have installed Oracle Data Integrator (ODI), and configured ODI for use by Oracle Fusion applications, you can load data to the staging tables by scheduling the Perform Data Load to Staging Tables process, PerformOdiSatagingLoad. To use this process, you must perform these steps and understand these details:

- Steps to use the Perform Data Load to Staging Tables process
- Steps to manually prepare and update the required dat files
- Details regarding the Perform Data Load to Staging Tables process
- Steps to verify execution status after starting the Perform Data Load to Staging Tables process
- Details regarding verifying the Perform Data Load to Staging Tables process execution status
- List of interface ODI scenarios run for each business entity

Steps to Use the Perform Data Load to Staging Tables Process

The Perform Data Load to Staging Tables process invokes an ODI data load. To use this process, follow these steps:

1. Create a data file for each business entity for which you are extracting data from your external source system. The file type for the data files must be dat. Use the sample dat files provided on My Oracle Support as templates. The data in the files you create must conform to the exact formats provided in the sample files.
 - To obtain the sample dat files, see Oracle Fusion Order Promising Data Collections Sample ODI Data Files, document ID 1361518.1, on My Oracle Support <https://support.oracle.com>.
 - You can open the sample dat files in a spreadsheet tool to review the sample data. The sample data shows the minimum required fields for an entity.
2. Place the dat files in the host where the Supply Chain Management (SCM) ODI agent is installed. The dat files must be placed at this specific location: /tmp/ODI_IN.
 - The path for this location is configured for the SCM ODI Agent. The SCM ODI Agent is an ODI software agent that services ODI related client requests. More information about this agent can be found in the ODI product documentation.
 - After ODI is installed, you must use the ODI console to refresh the variables C_LAST_UPDATED_BY and C_CREATED_BY.
3. Schedule the Perform Data Load to Staging Tables, PerformOdiStagingLoad, process.

Steps to Manually Prepare and Update the Required dat Files

You can develop data extract programs to extract data from your external source systems and store the extracted data into the required dat files in the required format. To manually add data to the dat files, follow these steps:

1. Open the applicable dat file in a spreadsheet tool. When you open the file, you will be prompted to specify the delimiter.

Use the tilde character, ~ , for the delimiter.

2. Add any data records you want to upload to the staging tables into the spreadsheet. Data for date type columns must be in the DD-MON-YY date format.
3. Save the worksheet from the spreadsheet tool into a text file.
4. Use a text editor and replace spaces between columns with the tilde character.
5. Verify that every line terminates with a CR and LF (ASCII 000A & 000D respectively.)
6. Upload the dat file to the /tmp/ODI_IN directory where the SCM ODI agent is running. The location is seeded in the ODI topology. Upload (FTP) the dat file in binary mode only.

7. Review the file in vi after the FTP upload to detect junk characters and, if any, remove them.

Details Regarding the Perform Data Load to Staging Tables Process

The Perform Data Load to Staging Tables process invokes the ODI scenario MASTER_PACKAGE that internally invokes all four projects defined in ODI for collections. Each of these four projects invokes various interfaces. Data is loaded from flat files to staging tables for all the business objects enabled for Oracle Fusion 11.1.2.0.0 through Oracle Data Integrator.

The following are specific details for the process:

- Process Name: PerformOdiStagingLoad
- Process Display Name: Perform Data Load to Staging Tables
- Process Description: Collects planning data from flat files and loads to staging tables using Oracle Data Integrator.
- ODI Project Name: SCM_BulkImport
- ODI scenario Name: MASTER_PACKAGE
- SCM Scheduler: SCM_ESS_ODI_SCHEDULER
- Agent URL: your_host_name:your_port_no/oracleodiagent (substitute your host name and your port number)

Steps to Verify Execution Status after Starting the Perform Data Load to Staging Tables Process

To verify the execution status after starting the Perform Data Load to Staging Tables process, perform these steps:

1. The Perform Data Load to Staging Tables process does not log messages to the scheduled processes side. To check for a log message, query the Request_History table using this select statement:

```
Select * from fusion_ora_ess.request_history where requestid=  
<request_id>;
```

- Check the Status column for the overall execution status of the job and the Error_Warning_Detail column for a detailed error message, if any.
2. Check the ODI scenario execution status details in the ODI operator window. The scenario names are listed in the table in the List of Interface ODI Scenarios Run for Each Business Entity section of this document.
3. If log directories are accessible, check the following ODI logs for specific information on ODI scenario execution path:

```
/slot/emsYOUR_SLOT_NUMBER/apmgr/WLS/user_projects/domains/  
wls_appYOUR_SLOT_NUMBER/servers/YOUR_ODI_SERVER_NAME/logs
```

- Diagnostic: for any errors in execution
- Server: for all the logs specific to ODI console
- Agent: for scenario entry and exit and for session ID

Details Regarding Verifying the Perform Data Load to Staging Tables Process Execution Status

When verifying the Perform Data Load to Staging Table process, remember the following:

- No logs will be written at the scheduled processes side. Also, the session id for ODI scenario cannot be found at the scheduled processes side.
- When viewing the process status on the Scheduled Processes page, a Success status does not mean that all the data got into the staging tables successfully. The Success status only indicates that the scenario is launched successfully. Scenario status must be checked from ODI logs.
- You cannot determine the refresh_number generated by ODI for the current process run from the Scheduled Processes page. To obtain the refresh number, you must use this query to query the msc_coll_cycle_status table and check for the ODI collection_channel:

```
Select * from msc_coll_cycle_status order by refresh_number desc;
```

List of Interface ODI Scenarios Run for Each BusinessEntity

One or more interface ODI scenarios are run for each business entity. Each interface scenario maps to one entity. If any interface Scenario fails in ODI, that entity data is not collected to the staging tables. This table lists the business entities and the interface ODI scenarios run within each business entity.

Business Entity	Interface ODI Scenarios
Work-in-Process Requirements	WIP_COMP_DEMANDS_SCEN WIP_OP_RESOURCE_SCEN
Calendars	CALENDAR_SCEN CALENDAR_WORKDAYS_SCEN CALENDARDATES_SCEN CALENDAR_EXCEPTIONS_SCEN CALENDARSHIFTS_SCEN CALENDAR_PERIODSTARTDAYS_SCEN CALENDAR_WEEKSTARTDAY_SCEN CALENDAR_ASSIGNMENTS_SCEN
Demand Classes	DEMAND_CLASS_SCEN
Global Supplier Capacities	GLOBAL_SUP_CAPACITIES_SCEN
Interorganization Shipment Methods	SHIPMENT_METHODS_SCEN
Item Cost	ITEM_COST_SCEN
Item Substitutes	ITEM_SUBSTITUTES_SCEN
Item Suppliers (Approved Supplier List)	ITEM_SUPPLIERS_SCEN
On Hand	ONHAND_SCEN
Organizations	ORGANIZATIONS_SCEN

Purchase Orders and Requisitions	SUPPLY_INTRANSIT_SCEN PO_IN_RECEIVING_SCEN PO_SCEN PR_SCEN
Planned Order Supplies	PLANNEDORDERSUP_SCEN
Resources	RESOURCES_SCEN RESOURCE_CHANGE_SCEN RESOURCE_SHIFTS_SCEN RESOURCE_AVAILABILITY_SCEN
Routings	ROUTING_OPERATION_RESOURCES_SCEN ROUTINGS_SCEN ROUTING_OPERATIONS_SCEN
Sourcing Rules	SOURCING_ASSIGNMENTS_SCEN SOURCING_RULES_SCEN SOURCING_ASSIGNMENTSETS_SCEN SOURCING_RECEIPT_ORGS_SCEN SOURCING_SOURCE_ORGS_SCEN
Subinventories	SUB_INVENTORIES_SCEN
Trading Partners	TRADING_PARTNERS_SCEN TRADING_PARTNER_SITES_SCEN
Units of Measure	UOM_SCEN UOM_CONVERSION_SCEN UOM_CLASS_CONVERSION_SCEN
Work Order Supplies	WORKORDER_SUPPLY_SCEN

Parameters for the Perform Data Load Process: Points to Consider

To perform a data load from the data collection staging tables, you invoke the Perform Data Load from Staging Tables process. When you invoke the process, you provide values for the parameters used by the process

Parameters for the Perform Data Load from Staging Tables Process

When you perform an upload from the staging tables, you specify values for a set of parameters for the Perform Data Load from Staging Tables process including specifying Yes or No for each of the entities you can load. For the parameters that are not just entities to select, the table below explains the name of each parameter, the options for the parameter values, and the effect of each option.

Parameter Name	Parameter Options and Option Effects
Source System	Select from a list of source systems.
Collection Type	<ul style="list-style-type: none"> • Net change Data in the data repository is updated with the data uploaded from the staging tables. <ul style="list-style-type: none"> • Existing records are updated. For example, on hand is updated with current quantity. • New records are added to the data repository. For example, New purchase orders are added to the data repository. • Targeted Existing data in the data repository is deleted and replaced with the data uploaded from the staging tables. For example, a targeted data load for purchase orders will replace all existing purchase order data with the purchase order data from the staging tables.
Group Identifier	Leave blank or select from the list of collection cycle identifiers. Leave blank to load all staging table data for the selected collection entities. Select a specific collection cycle identifier to load data for that collection cycle only.
Regenerate Calendar Dates	<ul style="list-style-type: none"> • Yes You loaded calendar patterns into the staging tables so you need the concurrent process to generate and store individual dates to run. • No You loaded individual dates into the staging tables so you do not need the concurrent process to generate and store individual dates to run.
Regenerate Resource Availability	<ul style="list-style-type: none"> • Yes You loaded resource availability patterns into the staging tables so you need the concurrent process to generate and store individual dates to run. • No You loaded individual dates into the staging tables so you do not need the concurrent process to generate and store individual dates to run.

The parameters presented for the Perform Data Load from Staging Tables process also include a yes-or-no parameter for each of the entities you can collect

using the staging tables upload method. If you select yes for all of the entities, the data collections will be performed in the sequence necessary to avoid errors caused by data references from one entity being loaded to another entity being loaded.

Important

If you do not select yes for all of the entities, you need to plan your load sequences to avoid errors that could occur because one of the entities being loaded is referring to data in another entity not yet loaded. For more information, see the articles regarding order promising or data collections on My Oracle Support at <https://support.oracle.com>.

Collections Cycle Identifier: Explained

The collection cycle identifier is a unique number that identifies a specific data collection cycle, or occurrence. One cycle of a data collection covers the time required to collect the set of entities specified to be collected for a specific data collection method. The collection cycle identifier is then used in statistics regarding data collections, such as the Data Collection Summary report. The collection cycle identifier is also used for a parameter in various processes related to data collections, such as the Purge Staging Tables process and the Perform Data Load process.

This topic explains the population of the collection cycle identifier when you use collecting data from external source systems as follows:

- Web Service Uploads and the Collection Cycle Identifier
- Staging Tables Uploads and the Collection Cycle Identifier

Web Service Uploads and the Collection Cycle Identifier

When you use the Web service upload data collection method, a collection cycle identifier is included as part of the collected data. You can then use the collection cycle identifier to review statistics regarding the Web service collections, or to search for error and warning records written to the data collection staging tables.

Staging Table Uploads and the Collection Cycle Identifier

If you use the Oracle Data Integrator tool to load your extracted data into the data collections staging tables, a collection cycle identifier is created for each load session. Each record loaded into the staging table during the load session will include the collection cycle identifier for that session.

If you populate the data collection staging tables using a method other than the Oracle Data Integrator tool, you must follow these steps to populate the collection cycle identifier.

1. Groupid is to be populated in column refresh_number of each data collections staging table. In one cycle of loading data into the staging tables, the column should be populated with same value. Get the group id value as follows:

```
SELECT ...NEXTVAL FROM DUAL;
```

2. After a cycle loading data into the data collections staging tables, insert a row as follows into table `msc_cycle_status` for that cycle as follows:

```
INSERT INTO MSC_COLL_CYCLE_STATUS
(INSTANCE_CODE, INSTANCE_ID, REFRESH_NUMBER, PROC_PHASE, STATUS,
COLLECTION_CHANNEL, COLLECTION_MODE, CREATED_BY, CREATION_DATE,
LAST_UPDATED_BY, LAST_UPDATE_DATE)
SELECT a.instance_code, a.instance_id, :b1, 'DONE', 'NORMAL',
'LOAD_INTERFACE', 'OTHER', 'USER', SYSTIMESTAMP, USER, SYSTIMESTAMP
FROM msc_apps_instances a
WHERE a.instance_code= :b2 ;
:b1 is instance_code for which data is loaded
:b2 is the groupid value populated in column refresh_number in all
interface tables for this cycle
```

Collecting Calendars and Resource Availability: Points to Consider

When you collect calendars and net resource availability from external source systems, you decide whether to collect patterns or individual dates. Order promising requires individual calendar dates and individual resource availability dates to be stored in the order orchestration and planning data repository. If you collect calendar patterns or resource shift patterns, you must invoke processes to populate the order orchestration and planning data repository with the individual dates used by order promising.

You invoke the necessary processes by specifying the applicable parameters when you run data collections. The processes generate the individual dates by using the collected patterns as input. The processes then populate the order orchestration and planning data repository with the individual calendar dates and the individual resource availability dates.

Calendar Collections

When you collect calendars from external source systems, you decide whether to collect calendar patterns or individual calendar dates. Both methods for collecting data from external source systems, Web service upload and staging tables upload, include choosing whether individual calendar dates must be generated as follows:

- The Web service to upload to calendars includes a parameter to run the Generate Calendar Dates process.

You control whether the process will run. If the parameter is set to yes, then after the Web service upload completes, the process will be launched to generate and store individual calendar dates.

- The parameters for the Perform Data Load from Staging Tables process also include a parameter to run the Generate Calendar Dates process.

You control whether the process will run. If the parameter is set to yes, then after the load from staging tables completes, the process will be launched to generate and store individual calendar dates.

- In both scenarios, calendar data is not available while the Generate Calendar Dates process is running.

When you collect calendars from the Oracle Fusion system, the Generate Calendar Dates process is run automatically.

Restriction

Only calendar strings that are exactly equal to seven days are allowed. Calendar strings with lengths other than seven are not collected. Only calendars with Cycle = 7 should be used.

Resource Availability Collections

When you collect net resource availability from external source systems, you decide whether to collect resource shift patterns or individual resource availability dates. Both methods for collecting data from external source systems, Web service upload and staging tables upload, include specifying whether individual resource availability dates must be generated as follows:

- The Web service to upload to net resource availability includes a parameter to run the Generate Resource Availability process.

You control whether the process will run. If the parameter is set to Yes, then after the Web service upload completes, the process will be launched to generate and store individual resource availability dates.

- The parameters for the Perform Data Load from Staging Tables process also include a parameter to run the Generate Resource Availability process.

You control whether the process will run. If the parameter is set to Yes, then after the load from staging tables completes, the process will be launched to generate and store individual resource availability dates.

- In both scenarios, new resource availability data is not available while the Generate Resource Availability process is running.

You cannot collect net resource availability from the Oracle Fusion source system.

Parameters for the Perform Data Collection Process: Points to Consider

To perform a targeted data collection from the Oracle Fusion system, you use the Perform Data Collection process. When you invoke the process, you provide values for the parameters used by the process.

The Perform Data Collection Process

When you perform a targeted collection, you specify the Oracle Fusion source system to be collected from and the organization collection group to collect for. When you invoke the process, the parameters also include each of the fourteen entities you can collect from the Oracle Fusion source system with yes or no for the parameter options. The table below explains the other two parameters.

Parameter Name	Parameter Options
Source System	The source system presented for selection is determined by what system has been defined as the Oracle Fusion source system when the manage source systems task was performed.
Organization Collection Group	The organization collection groups presented for selection are determined by what organization groups were defined when the manage source systems task was performed for the selected source system.

The parameters presented also include a yes-or-no parameter for each of the entities you can collect. If you select yes for all of the entities, the data collections will be performed in the sequence necessary to avoid errors caused by data references from one entity being loaded to another entity being loaded.

Important

If you do not select yes for all of your entities, you need to plan your load sequences to avoid errors that could occur because one of the entities being loaded is referring to data in another entity not yet loaded. For more information, see the articles regarding order promising or data collections on My Oracle Support at <https://support.oracle.com>.

Organization Collection Group: Explained

When you perform a targeted collection from the Oracle Fusion source system, you use an organization collection group to contain the collections processing to only the organizations with data that is needed for the order orchestration and planning data repository. Organization collection groups limit targeted collections from the Oracle Fusion source system to a specific set of organizations.

You perform the following actions for organization collection groups:

- Define an organization collection group.
- Use an organization collection group.

Define an Organization Collection Group

You define organization groups when managing source systems for the source system where the version equals Oracle Fusion. For each organization in the organization list for the Oracle Fusion source system, you can specify an organization group. You can specify the same organization group for many organizations.

Use an Organization Collection Group

You use an organization collection group when you perform a targeted collection from the Oracle Fusion source system and you want to contain the collections processing to a specific set of organizations. You specify which organization

group to collect data from by selecting from the list of organization groups defined for the Oracle Fusion source system. Data will only be collected from the organizations in the organization group you specified.

For example, if only certain distribution centers in your Oracle Fusion source system are to be considered for shipments to your customers by the order promising and order orchestration processes, you could create a DC123 organization group and assign the applicable distribution centers to the DC123 organization group when managing source systems. When you perform a targeted collection for the Oracle Fusion source system, you could select DC123 for the organization collection group.

Review Planning Collected Data

Data Collections Daily Monitoring: Explained

When you manage the data collection processes, you use the Process Statistics report and the Data Collection Summary report to routinely monitor your collections. When error records are reported, you query the data staging tables for further details regarding the error records. You can also review most of your collected data using the review collected data pages.

The following information sources are available for you to monitor data collections:

- Process Statistics report
- Data Collection Summary report
- Review collected data pages
- Staging table queries

Process Statistics Report

You view the Process Statistics report to monitor summary of statistic for the daily collections activity for each of your source systems. This report is available on the **Actions** menu when managing data collection processes for either the continuous collection publish process or the collections destination server. The day starts at 00:00 based on the time zone of the collection server.

For the Oracle Fusion source system, statistics are provided for both the continuous collection and the targeted collection data collection methods. For each external source system, statistics are provided for the Web service upload and for the staging tables upload data collection methods. The following statistics are provided in the Process Statistics report:

- Number of collection cycles for the current day
- Average cycle time in seconds
- Average number of records
- Average number of data errors

Note

The process statistics provide summary information, and are not intended for detailed analysis of the collections steps. Use the Oracle Enterprise Scheduler Service log files for detailed analysis.

Data Collection Summaries

You view the Data Collection Summary report to monitor statistics regarding the data collection cycles for each of your source systems. The summary report shows last the results of the last 20 cycles of all collection types. This report is available on the Action menu when managing data collection processes for the collections destination server.

The Data Collection Summary report provides information for each source system. If a source system was not subject to a data collection cycle for the period covered by the summary, an entry in the report states that there are no cycles in the cycle history for that source system. For each source system that was subject to a data collection cycle for the period covered by the summary, the following information is provided for each data collection method and collected entity value combination:

- The data collection method
- The collection cycle number
- The entity collected and, for that entity, the number of records collected, the number of records with data errors, and collection duration
- Time started
- Time ended

Review Collected Data Pages

You can review most of your collected data by using the Review Planning Collected Data page or the Review Order Orchestration Collected Data page. Both pages include a list of entities from which you select to specify the entity for which you want to review collected data. The list of entities is the same on both pages. Most of the entities listed on the review collected data pages are identical to the entities you select from when you run collections, but there are a few differences.

Some of the entities on the list of entities you select from when you review collected data are a combination or a decomposition of the entities you select from when you run collections. For example, the Currencies data collection entity is decomposed into the Currencies entity and the Currency Conversions entity on the review collected data pages. For another example, the Supplies entity on the review collected data pages is a combination of data collection entities including the On Hand entity and the Purchase Orders and Requisitions entity.

A few of the data collection entities cannot be reviewed from the review collected data pages. The data collection entities that are not available for review on the review collected data pages are Resources, Resource Availability, Routings, Work-in-Process Resource Requirements, and Customer Item Relationships.

Staging Table Queries

If errors or warnings have been encountered during data collections, you can submit queries against the staging tables to examine the applicable records. For more information regarding the staging tables and staging table columns, see the articles regarding order promising or data collections on My Oracle Support at <https://support.oracle.com>.

Errors and Warnings When Collecting Data from External Source Systems: How They Are Handled

When you are collecting data from external source systems, the data collection processes perform many data validation checks. If the data validations fail with errors or warnings, the steps taken by the data collection processes vary slightly depending upon whether the Web service upload data collection method or the staging tables upload data collection method is used.

In both cases, records where errors are found are not loaded into the order orchestration and planning data repository. Instead records are loaded into, or remain in, the applicable staging tables with an appropriate error message. Records where only warnings are found are loaded to the data repository, and records are loaded into, or remain in, the applicable staging tables with an appropriate warning message.

Settings That Affect Error Handling When Collecting Data from External Source Systems

The handling of errors and warnings encountered when the data collection processes validate data during collections from external source systems depends upon which data collection method is used, Web service upload or staging tables upload.

How Errors and Warnings Are Handled

When you are running data collections using the Web services method, records without errors or warnings are loaded into the data repository. For records with errors or warnings the following error and warning handling steps occur when using the Web services method:

- **Warnings:** Records with warnings are fixed automatically, then loaded into the data repository and copied into the applicable staging tables with the appropriate warning message.
- **Errors:** Records are loaded to the applicable staging tables instead of the data repository and are marked with the appropriate error message. When there is an error due to missing mandatory fields, in cases where possible, the collections process will retry the record. After several unsuccessful retry attempts, the record will be marked as error. In some cases, retrying is not an option, and the record will be immediately marked as an error.

When you are running data collections using the staging tables upload method, the following error and warning handling steps occur:

- **Warnings:** Records are loaded into the data repository and remain in the staging tables with the appropriate warning message. The message is associated with the record in the data repository.
- **Errors:** When there is an error due to missing mandatory fields, in cases where possible, the collections process will retry the record. After several unsuccessful retry attempts, the record will be marked as error. In some cases, retrying is not an option, and the record will be immediately marked as an error.

Error Handling Example

When a Planned Order Supplies record is collected, many validations occur for which an error is recorded if the validation fails.

For example, the supplier name is validated against the suppliers data in the order orchestration and planning data repository. If the supplier name is not found, the validation fails with an error condition, and the following steps occur:

- The Planned Order Supplies record is not loaded into the data repository.
- The Planned Order Supplies record is loaded into the applicable staging table, or remains in the applicable staging table, with an error message stating invalid supplier or invalid supplier site.

Warning Handling Example

When a Planned Order Supplies record is collected, many validations occur for which a warning is recorded if the validation fails.

For example, the Firm-Planned-Type value in the record is validated to verify that the value is either 1 for firm or 2 for not firm. If the validation fails, the failure is handled as a warning, and the following steps occur:

- The Planned Order Supplies record is loaded into the data repository with the Firm-Planned-Type value defaulted to 2 for not firm.
- The Planned Order Supplies record is also loaded into the applicable staging table, or remains in the applicable staging table, with a warning message stating invalid firm planned type.

Purge Collected Data Processes: Points to Consider

You use the Purge Data Repository Tables process to delete all collected data from the order orchestration and planning data repository that was collected from a specific source system. You use the Purge Staging Tables process to remove data that you no longer need in the data collections staging tables.

The Purge Data Repository Tables Process

You use the Purge Data Repository process to delete all data for a source system from the order orchestration and planning data repository. The process enables you to delete data for a specific source system. You typically use the Purge Data Repository process when one of your source systems becomes obsolete, or when you decide to do a complete data refresh for a set of collection entities.

The Purge Data Repository process has only two parameters, both of which are mandatory. This table explains the two parameters.

Parameter Name	Parameter Options
Source System	Select a source system for the list of source systems. All data for the selected system will be deleted from the data repository.
Purge Global Entities	Yes or No If you select yes, in addition to the applicable data being deleted for the source-specific entities, all data from global entities will also be deleted. If you select no, data will be deleted from the source-specific entities only.

The Purge Staging Tables Process

You use the Purge Staging Tables process to delete data from the data collection staging tables.

The following table explains the parameters you specify when you run the Purge Staging Tables process. In addition to the five parameters explained below, you specify yes or no for each of the twenty-five data collection entities.

Parameter Name	Parameter Options
Source System	Select a source system for the list of source systems. Data will be deleted for this source system only.
Record Type	The record type specifies which type of records to purge as follows: <ul style="list-style-type: none"> • Error Purge only error records. • Warning Purge only warning records. • Retry Purge only records marked as retry. • Complete Purge only records that have been successfully processed and data stored in the data repository. • All Purge all records.
Collection Cycle ID	Specify a value for the collection cycle identifier to purge data for a specific collection cycle only, or leave blank.

From Date Collected	Specify a date to purge data from that date only, or leave blank.
To Date Collected	Specify a date to purge data up to that date only, or leave blank.

FAQs for Review Planning Collected Data

What's an order orchestration reference object?

One of the objects in the set of objects used by the orchestration processes to determine the meaning and descriptions for names or codes, such as payment terms names, freight-on-board codes, and mode-of-transport codes.

The sales order data passed to the orchestration processes contains the names or codes, but the processes need to display the meanings or descriptions. The data to determine the meanings or descriptions for the names or codes must be collected into the order orchestration and planning data repository.

For example, sales order information is passed to the Order Orchestration processes containing a freight-on-board code equal to 65, and the order orchestration and planning data repository contains a record with freight-on-board code equal to 65. The processes use the matching codes to determine that the freight-on-board code meaning is equal to Origin, and the description is equal to Vendors responsibility.

Tip

For the full list of order orchestration reference objects, review collected data for the order orchestration reference objects, and view the list of values for the **Lookup Type** field.

Manage Sourcing Rules and Bills of Distribution

Sourcing Rules and Bills of Distribution: Explained

To define the sources of supply for your supply chains and to define your date-effective sourcing strategies, create sourcing rules and bills of distribution. Within each sourcing rule or bill of distribution, you define one or more supply sources and a combination of rankings and quantity-based sourcing specifications for each source to define priorities across the supply sources. For each source, you also select one of three source types, and you specify the value for the attributes applicable to the selected source type.

This table lists the three replenishment source types, the definition of the source type, and the attributes to specify for each source type.

Source Type	Source Type Definition	Attributes to Specify
Buy from	Sourced from an external supplier.	Specify the supplier and supplier site.

Make at	Sourced from an internal organization that manufactures the item.	Specify the manufacturing organization.
Transfer from	Sourced through an interorganization transfer.	Specify the organization from which items will be transferred.

Note

When you create sourcing rules and bills of distribution, you specify how you will replenish items. You do not specify what items that you will replenish. To specify which sourcing rules or bills of distribution that you will use to replenish what items, you create assignment sets.

You define the following aspects of sourcing rules and bills of distribution to define your sources of supply and your sourcing strategies:

- Global sourcing rules
- Local sourcing rules
- Bills of distribution
- Effectivity dates
- Source ranks, quantity-based sourcing specifications, and allocation percentages

Tip

When first designing your sourcing rules and bills of distribution, start by envisioning your assignment set. Determine what set of global sourcing rules, local sourcing rules, bills of distribution, or combinations of rules and bills that you need to implement your assignment set while minimizing the number of rules or bills to maintain. For example, you may be able to define a global sourcing rule in such a way that you will need only a few local sourcing rules to assign for exceptions to the global rule.

Global Sourcing Rules

Global sourcing rules can specify two of the source types: the buy-from or transfer-from source types. Any organization can potentially replenish items by buying from any of the suppliers specified in the buy-from sources, or transferring from any of the organizations specified in the transfer-from sources. For example, if you create a global sourcing rule with a buy-from source with Super Supply Company specified for the supplier, any of your organizations can potentially buy from Super Supply Company.

If you have a source that is applicable to most of your organizations, create a global sourcing rule for that source and local sourcing rules for the organizations for which the source is not applicable. For example, if there are 20 organizations in your company, and 19 of the organizations transfer supply from the Munich organization, create a global sourcing rule specifying transfer-from the Munich organization, and create a local sourcing rule specifying where the Munich organization gets supply from.

Local Sourcing Rules

Local sourcing rules can specify all three source types. Because a local sourcing rule is applicable to one, and only one, organization, you specify which organization the rule is being created for when you create the rule. The replenishment sources defined in the rule are applicable only to the organization for which the rule was created. For example, if you create a local sourcing rule with M1 as the organization for which the rule is being created, and you add a buy-from source to the rule with XYZ Supply Company specified for the supplier, and you have no other sourcing rules or bills of distribution with XYZ Company specified for the supplier, then only the M1 organization can buy from XYZ Supply Company.

Bills of Distribution

If you have designed multiple local sourcing rules with material flowing through three or more organizations, you can choose to create one bill of distribution to implement the sources instead of creating multiple local sourcing rules. Choosing to create a bill of distribution instead of sourcing rules is a personal or organizational preference. Any scenario that you can implement by creating a bill of distribution, you can also implement by creating multiple local sourcing rules.

For example, the following sourcing scenario could be implemented by three local sourcing rules or one bill of distribution:

- Organization M1 sources items by purchasing from a supplier, XYZ Supply.
- Organization M2 sources items by transferring from M1.
- Organization M3 sources items by transferring from M2.

Effectivity Dates

Use sourcing effectivity dates to modify sourcing rules and bills of distribution when sources change, such as a new supplier contract is established or a manufacturing facility is shut down. Each rule or bill can have multiple, non-overlapping ranges of effectivity start dates and end dates, with a different set of sources specified for each range. For example, if you have a sourcing rule that currently specifies a buy-from source with Acme Supplier specified for the supplier, but your company has decided to start buying from Winter Widgets instead, you would modify the sourcing rule by specifying the applicable end date, the date you will no longer buy from Acme Supplier, for the current effectivity date range. You add a new effectivity date range, specifying the date when you will start buying from Winter Widgets for the start date, and then you add a buy-from source for the new effectivity date range with Winter Widgets specified for the supplier.

Source Ranks, Quantity-Based Sourcing Specifications, and Allocation Percentages

For each source in a sourcing rule or bill of distribution, you designate a rank to specify the order in which the sources within the rule or bill will be considered by order promising when the rule or bill is applied during a supply chain

availability search. The source with the lowest number rank will be considered first, and the source with the highest number rank will be considered last. If your sourcing strategy includes using specific sources for specific quantities, you designate a from quantity, a less-than quantity, or both, for one or more sources.

Note

Because sourcing rules collected from external source systems may include split allocations for planning purposes, there may be multiple sources with the same rank and quantity range, but the allocation percentages must add up to 100 percent. The Order Promising process does not split the desired quantity when checking for availability.

The Order Promising process checks the source with the highest allocation percent first within a group of sources with the same rank. If the source with the highest allocation percent has enough supply, that source is used for the entire requested quantity. If the source with the highest allocation percent does not have enough supply, then the source with the next highest allocation percent will be checked for the entire quantity. Because split allocations are not applicable to order promising sourcing strategies, the examples provided here do not include split allocations.

The following table is an example of a sourcing rule with three ranks. Quantity-based sourcing is not being used in this example. If a supply chain search is conducted using this rule, order promising checks if organization M2 can make the desired quantity first. If organization M2 cannot make the desired quantity, order promising will then check if there is enough quantity at organization V1 for an interorganization transfer. If there is not enough quantity at organization V1, then order promising will check if the desired quantity can be bought from supplier Winter Widgets.

Replenishment Source and Applicable Attribute Value	Rank	Allocation Percent
Make at manufacturing organization M2	1	100
Transfer from organization V1	2	100
Buy from supplier Winter Widgets	3	100

Defining Quantity-Based Sourcing for Multiple Sources: Example

This example illustrates how to define sourcing rules to implement sourcing requirements with quantity-based sourcing specified in the requirements.

Scenario

You are defining the sources for a set of business requirements that initially include quantity ranges for two suppliers. The requirements change to include a third quantity range and a third supplier.

Quantity-Based Sourcing Specifications

Your business initially defines the following sourcing requirements:

- For quantities less than 100, buy from Supplier A.
- For quantities greater than or equal to 100, buy from Supplier B.

Your business adds a new supplier, Supplier C. Your business now defines the following sourcing requirements:

- For quantities less than 100, buy from Supplier A, if Supplier A has enough quantity.
- For quantities greater than or equal to 100, but less than 200, buy from Supplier B, if Supplier B has enough quantity.
- For quantities greater than or equal to 200, buy from Supplier C.
- If Supplier A does not have enough supply for a quantity less than 100, or Supplier B does not have enough supply for a quantity between 100 and 199, buy from Supplier C for these quantities.

Analysis

First, analyze your sourcing requirements to determine how many different sourcing rules you need to create to implement your sourcing requirements. The requirements specified above can be defined within one sourcing rule. After determining how many sourcing rules to define, determine how many sources must be defined for each sourcing rule. First analyze how many replenishment source types have been specified in the requirements. All of the requirements above are for buy-from-a-supplier replenishment source types. Next, analyze how to define the From Quantity, Less Than Quantity, and Rank attributes as needed to implement your sourcing requirements.

For the requirements as initially stated, define two sources with the following values for the Source Type, Supplier, From Quantity, Less Than Quantity, Allocation, and Rank attributes:

- Source Type equals Buy from, Supplier equals Supplier A, Less Than Quantity equals 100, Allocation Percent Equals 100, and Rank equals 1
 - You do not need to specify a value for the From Quantity attribute because the source applies for any quantity less than 100.
- Source Type equals Buy from, Supplier equals Supplier B, From Quantity equals 100, Allocation Percent Equals 100, and Rank equals 1.
 - You do not need to specify a value for the Less Than Quantity attribute because the source applies for any quantity greater than or equal to 100.

For the requirements after the third supplier is added, edit the buy-from-Supplier-B source and add additional sources for Supplier C to define the four sources with the following values for the Source Type, Supplier, From Quantity, Less Than Quantity, Allocation, and Rank attributes:

- Source Type equals Buy from, Supplier A, Less Than Quantity equals 100, Allocation Percent Equals 100, and Rank equals 1.
 - You do not need to specify a value for the From Quantity attribute because the source applies for any quantity less than 100.

- Source Type equals Buy from, Supplier B, From Quantity equals 100, Less Than Quantity equals 200, Allocation Percent Equals 100, and Rank equals 1.
- Source Type equals Buy from, Supplier C, From Quantity equals 200, Allocation Percent Equals 100, and Rank equals 1.
 - You do not need to specify a value for the Less Than Quantity attribute because the source applies for any quantity greater than or equal to 200.
- Source Type equals Buy from, Supplier C, and Rank equals 2.
 - You do not need to specify a value for the From Quantity attribute or Less Than Quantity attribute because there is no minimum or maximum value in this case.

This table lists the two sources you define to implement the following sourcing requirements:

- Check Supplier A for order quantities less than 100.
- Check Supplier B for order quantities greater than or equal to 100.

Type	Supplier	From Quantity	Less Than Quantity	Quantity Unit of Measure	Allocation Percent	Rank
Buy from	A		100	Each	100	1
Buy from	B	100		Each	100	1

This table lists the four sources you define to implement the following sourcing requirements:

- Check Supplier A for orders for quantities less than 100.
- Check supplier B for quantities greater than or equal to 100 and less than 200.
- Check supplier C for quantities greater than 200.
- Check supplier C for quantities less than 200 when Supplier A or Supplier B do not have the desired quantity.

Type	Supplier	From Quantity	Less Than Quantity	Quantity Unit of Measure	Allocation Percent	Rank
Buy from	A	1	100	Each	100	1
Buy from	B	100	200	Each	100	1
Buy from	C	200		Each	100	1
Buy from	C				100	2

Manage Assignment Sets

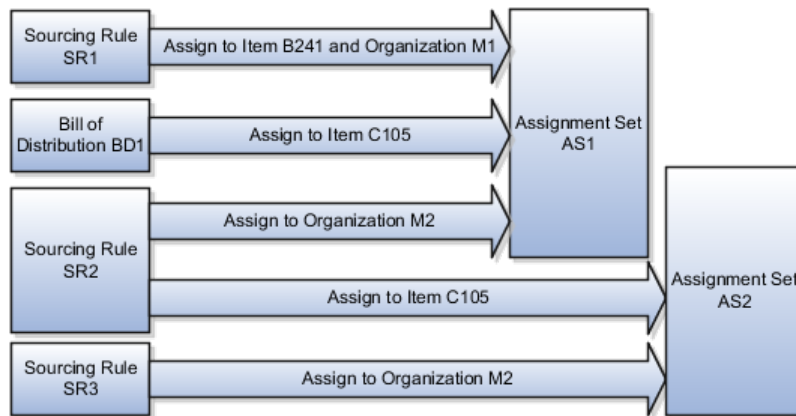
Assignment Sets, Sourcing Rules, and Bills of Distribution: How They Work Together

You create assignment sets to implement the supply chain networks for your sourcing strategies. You implement your supply chain network by selecting the appropriate sourcing assignment level when you assign a sourcing rule or bill of distribution to an assignment set. You create alternative assignment sets, with different sourcing assignments, to model alternative supply chains.

The following figure shows an example where three sourcing rules and one bill of distribution are assigned to two assignment sets:

- The first sourcing rule, SR1, is assigned to the first assignment set, AS1, at the item and organization assignment level for item B241 and organization M1.
- The bill of distribution, BD1, is assigned to the first assignment set, AS1, at the item assignment level for item C105.
- The second sourcing rule, SR2, is assigned to the first assignment set, AS1, at the organization assignment level for organization M2.
- The second sourcing rule, SR2, is also assigned to the second assignment set, AS2, but it is assigned to AS2 at the item assignment level for item C105.
- The third sourcing rule, SR3, is assigned to the second assignment set AS2, at the organization assignment level for organization M2.

When the supply chain network implemented by assignment set AS2 is followed, Item C105 is replenished according to the sourcing means specified in the sourcing rule SR2. When the supply chain network implemented by assignment set AS1 is followed, Item C105 is replenished according to the sourcing means specified in the bill of distribution BD1.



Assigning Sourcing Rules or Bills of Distribution to Assignment Sets

When you create sourcing rules and bills of distribution, you create descriptions of the means by which you replenish items, but you do not associate these means

with any specific items. You create assignment sets to define your supply chain sourcing and transfer links by assigning sourcing rules and bills of distribution to specific items, customers, organizations, categories, demand classes, or regions. For each sourcing assignment within an assignment set, you select the applicable sourcing assignment level to implement the scope of the sourcing rule or bill of distribution for the specific sourcing assignment.

When you add new replenishment sources, change your strategies for using your existing sources, or you delete replenishment sources, you edit existing assignment sets, or create new assignment sets, to incorporate these changes into your supply chains. When you edit assignment sets, you add new sourcing assignments to the assignment set, delete existing sourcing assignments from the assignment set, or make changes to the assignment level and assignment attributes for existing sourcing assignments. You edit assignment sets on the Edit Assignment Set page, or in a worksheet by choosing to edit in worksheet while on the Manage Assignment Sets or Edit Assignment Set pages.

Sourcing Assignment Levels: Explained

When you design an assignment set, you determine the sourcing assignment level for each sourcing assignment contained within the assignment set. To implement well-designed assignment sets, you must know which sourcing assignment levels take precedence over which other sourcing assignment levels.

Two aspects to understand regarding sourcing assignment levels are:

- The sourcing assignment levels and their levels of granularity
- Sourcing demand types and the sourcing assignment levels

Sourcing Assignment Levels and Their Levels of Granularity

To determine which sourcing assignments to include in an assignment set, you need to know which assignment levels override which assignment levels. An assignment level that is more granular overrides an assignment level that is less granular.

For example, the Item and Customer and Customer Site assignment level is more granular than the Item and Customer assignment level. If a customer has 12 customer sites, and your sourcing strategy is the same for a specific item at 11 of the 12 customer sites, you only need to add these two sourcing assignments to your assignment set to implement this aspect of your sourcing strategy:

- A sourcing assignment at the Item and Customer assignment level to implement which sourcing rule or bill of distribution is applicable for orders placed for the item by the customer at 11 of the customer sites.
- A sourcing assignment at the Item and Customer and Customer Site assignment level to implement which sourcing rule or bill of distribution is applicable for orders placed for the item by the customer at the twelfth customer site.

If an order for the item is received for the customer at the twelfth customer site, then the sourcing rule or bill of distribution assigned at the Item and Customer

and Customer Site level will be applied. If an order for the item is received for the customer for any of the other eleven sites, then the sourcing rule or bill of distribution assigned at the Item and Customer assignment level will be applied.

The sourcing assignment levels, listed most granular to least granular, are:

- Item and customer and customer site: Applies to a specific item for a specific customer at a specific customer site.
- Item and customer: Applies to a specific item for a specific customer at all of the customer's sites.
- Item and demand class: Applies to a specific item in a specific demand class.
- Item and region: Applies to a specific item in a specific region or zone.
- Item and organization: Applies to a specific item at a specific organization.
- Category and customer and customer site: Applies to all items in specific item category for a specific customer at a specific customer site.
- Category and customer: Applies to all items in specific item category for a specific customer at all of the customer's sites.
- Category and demand class: Applies to all items in a specific item category for a specific demand class.
- Category and organization: Applies to items in a specific item category at a specific organization.
- Item: Applies to a specific item in all regions, in all demand classes, and for all customers and all organizations.
- Category and region: Applies to all items in a specific item category for a specific region.
- Category: Applies to all items in a specific item category in all regions, in all demand classes, and for all customers and all organizations.
- Customer and customer site: Applies to a specific customer at a specific customer site for all items.
- Customer: Applies a specific customer at all of the customer's sites for all items.
- Demand class: Applies to all a specific demand class for all customers and all items.
- Region: Applies to a specific region or zone for all demand classes, all customers, and all items.
- Organization: Applies to a specific organization for all categories and all items.
- Global: Applies to all regions and zones, all demand classes, all customers, all items, and all organizations.

Note

The assignment levels that include category are available only if a category set has been defined for the Sourcing Rule Category Set profile option.

Sourcing Demand Types and the Sourcing Assignment Levels

When you create an assignment set, all assignment levels are applicable. When sourcing logic determines which sourcing assignment to use, the type of sourcing need determines what attribute values have been provided, which determines which assignment levels are considered.

Demand for sales orders or forecasts sourcing, also known as independent demand, specifies a value for one or more of the following attributes: item, customer, customer site, demand class. Sales orders always specify item, customer, and customer site. The postal code included in a customer site is used to derive the region. Therefore, for independent-demand sourcing the sourcing logic will consider sourcing assignments where the assignment level includes customer site, customer, item, demand class, or region. A sourcing assignment at the global assignment level will also be considered.

Organization demand specifies a value for the item. The category value is derived from the category the item belongs to. The organization the demand is for defines the organization value. Therefore, for organization-demand sourcing the sourcing logic will consider sourcing assignments where the assignment level includes item, category, or organization. A sourcing assignment at the global assignment level will also be considered.

Note

When sourcing logic is determining where to get the supply from for a specific independent demand, such as the demand specified by a fulfillment line, the answer may be to source it from an organization that doesn't have the supply on hand. At that point, the sourcing logic will use the assignment levels applicable to organization demand to determine how to source the supply for that organization.

Tip

If you are checking the availability for fulfillment line, and you are viewing the pegging tree presented when you view the details of an availability option, you can see the supply chain followed to determine how to source the fulfillment line.

Assignment Set Sourcing Hierarchy: How It Determines Which Sourcing Rule Is Used

The sourcing assignment levels that you select when you create sourcing assignments in an assignment set formulate a sourcing hierarchy for that assignment set. Order promising uses the sourcing hierarchy to determine which sourcing rule or bill of distribution to follow to find a source for a specific

item. Order promising always uses the most specific sourcing rule or bill of distribution that is applicable in the hierarchy.

Note

When order promising conducts a supply chain search, a profile option , the Default Order Promising Assignment Set profile option, designates which assignment set will be applied. Order promising uses the sourcing hierarchy to determine which sourcing rule or bill of distribution to follow from the rules or bills within the designated assignment set.

Settings That Affect the Sourcing Hierarchy

The position of a sourcing rule or a bill of distribution in the sourcing hierarchy is determined by these two factors:

- The assignment level at which you assigned the sourcing rule or bill of distribution to the assignment set.
- The rule or bill type which can be global sourcing rule, local sourcing rule, bill of distribution, or source organization. Source organization is the type used to designate when the set of item attribute values is what determines the source instead of a sourcing rule or bill of distribution.

Tip

Understanding and using the power of the sourcing hierarchy in an assignment set can make the designing and managing of sourcing relationships easier.

For example, if a plant initially receives all items belonging to a specific item category, such as the Fasteners item category, from Supplier A, then the sourcing rule to buy from Supplier A can be assigned at the Category assignment level for the Fastener item category.

If you then determine that a specific fastener is to be sourced from a different supplier, Supplier B for example, then you can assign a different sourcing rule to buy from Supplier B at the item level for the specific fastener. The detailed-to-general hierarchy determines that the specific fastener will be sourced from Supplier B, while all other fasteners are still sourced from Supplier A.

How the Sourcing Hierarchy Determines Which Rule Is Used

The sourcing hierarchy can be envisioned as a detailed-to-general table where each row in the table is a combination of assignment level and rule type. Each row in the hierarchy is more specific than the row below it. The topmost row, the row where a sourcing rule is assigned at the item and customer and customer site assignment level, is the most specific row. The bottommost row, the row where a global sourcing rule is assigned at the global assignment level, is the most general row. You use the sourcing hierarchy to answer which sourcing rule, bill of distribution, or set of item attribute values will be used to find a source for a specific combination of values of these four criteria:

- Assignment set

- Date
- Organization
- Item

For the sourcing rules and bills of distribution within the assignment set where the effective date of the sourcing assignment meets the date criteria, each rule or bill is associated with a specific row in the sourcing hierarchy. The sourcing assignment attribute values, such as the item value, determine which of the rules, bills, and set of item attributes are applicable to the specific criteria set. Multiple rules, bills, or item attributes can be applicable; therefore, multiple rows can be applicable. The rule, bill, or set of item attributes associated with the highest row in the hierarchy is the rule, bill, or set of item attributes that will be followed to determine the source.

From the Manage Assignment Sets page, you can select the **View Sourcing Hierarchy** button to view a table containing rows of the sourcing hierarchy. The most specific, most granular, row is the top row. The least specific, least granular row, is the bottom row.

Assignment Level	Sourcing Rule Type
Item and organization	Sourcing rule
Item and organization	Source Organization
Category and organization	Sourcing Rule
Item	Bill of Distribution
Item	Sourcing rule
Category	Bill of Distribution
Category	Sourcing Rule
Organization	Sourcing Rule
Organization	Source Organization
Global	Bill of Distribution
Global	Sourcing rule

Tip

You can view the sourcing hierarchy and initiate a search to ask "Where does this organization get this item on this date?" If you need to analyze why the order promising process returned results that were different than what you expected, you can view and search the sourcing hierarchy to determine which sourcing rule would be used for your set of criteria.

Editing an Assignment Set Within a Spreadsheet: Explained

When managing or editing assignment sets, you use the **Edit in Spreadsheet** button to use a spreadsheet to add, edit, or delete the sourcing rule or bill of distribution assignments for an assignment set. If you are managing assignment

sets, you must select an assignment set before you can choose to edit in spreadsheet.

Manage Global Order Promising Profile Options

Oracle Fusion Global Order Promising Profile Options: Critical Choices

Set profile options to specify the following for Oracle Fusion Global Order Promising:

- The sourcing assignment set and lead time multiplier used by the Check Availability process, as well as whether the Check Availability process can invoke external order promising web services
- The number of minutes that the results from the Check Availability process remain valid on the Check Availability page, as well as whether the check availability page displays analytics
- The number of default display days for the Review Supply Availability page, as well as the organization calendar to be used for supply buckets in the Supply Availability report
- The category set used when assignment sets are created

Check Availability Process

This table lists the profile options that affect the Check Availability process. If the profile option does not have a default value, the Default Value column in the table is blank.

Profile Option Display Name	Default Value	Effect
Order Promising Sourcing Assignment Set		Defines which sourcing assignment set will be used by the supply allocation and check availability processes
Supplier Capacity Accumulation Lead Time Multiplier	1	Defines the multiplier of the approved supplier list lead time to be used to determine the date when to begin the accumulation of supplier capacity
External ATP Web Service Enabled	No	If enabled, allows the Check Availability process to invoke external order promising web services

Check Availability Page

This table lists the profile options that affect the Check Availability page.

Profile Option Display Name	Default Value	Effect
Timeout for Check Availability Results	10	Sets the number of minutes that the results returned by the Check Availability process will remain valid on the Check Availability page
Analytics for Check Availability Page Enabled	Yes	If enabled, the Check Availability page will display analytics
Fulfillment Line Distribution Analytic Days for First Date Range	2	Sets the number of days for the first lateness range in the Fulfillment Line Distribution analytic
Fulfillment Line Distribution Analytic Days for Second Date Range	7	Sets the number of days for the second lateness range in the Fulfillment Line Distribution Analytic
Fulfillment Line Distribution Analytic Days for Third Date Range	14	Sets the number of days for the third lateness range in the Fulfillment Line Distribution Analytic

Review Supply Availability Page and Supply Availability Report

This table lists the profile options that affect the Review Supply Availability page and the Supply Availability report. If the profile option does not have a default value, the Default Value column in the table is blank.

Profile Option Display Name	Default Value	Effect
Default Display Days in Review Supply Availability Page	21	Sets the number of horizon days for the Review Supply Availability page if end date was not entered on the ATP Check Availability page
Organization Calendar for Supply Buckets in Supply Availability Report		Defines the organization calendar to use for the weekly and period supply buckets in the Supply Availability report

Assignment Set Assignment Level Selection

This table lists the Sourcing Rule Category Set profile option. There is no default value for the Sourcing Rule Category Set profile option. You must define a value for the Sourcing Rule Category Set profile option to have the assignment levels that include category available as choices for assignment level when creating assignment sets.

Profile Option Display Name	Effect
Sourcing Rule Category Set	Determines which category set is used when defining assignment sets

Manage Order Promising Rules

ATP Rules, Allocation Rules, and Sourcing Rules: How They Work Together

You create available-to-promise (ATP) rules and allocation rules to define how the items on fulfillment lines are promised by the order promising engine. ATP rules and allocation rules enable you to govern the behavior of the order promising engine. You can configure different types of rules and assign them in various ways to items and organizations to enable you to get different behavior for different items. You create sourcing rules, and assign the sourcing rules to assignment sets, to define your supply sources and supply chains used when order promising conducts a supply chain availability search.

ATP Rules

Oracle Fusion Global Order Promising provides an order promising engine with flexible promising process logic that you can direct per your business requirements through your definition of ATP rules. For example, constrained and high value items can be promised using supply chain availability search, while low value items can be promised by assuming infinite supply. The following are key points regarding ATP rules:

- ATP rules define order promising behavior by enabling you to specify:
 - The promising mode to be used by the order promising engine to determine how supply availability is considered
 - The supply and demand source types to be considered during promising
 - The usage of advanced promising features, such as capable to promise and profitable to promise
- ATP rules can be assigned flexibly to items, item organization, organization, or categories.
- By assigning different ATP rules to different items, fulfillment lines for various items can be promised in various ways.

Allocation Rules

Allocation rules enable you to specify portions of supply by demand classes to ensure high service levels for specific demand classes. For example, 80 percent

of supplies can be allocated to satisfy high priority demand classes. Allocation rules are applicable only when promising results are being determined through a supply chain availability search. If order promising is using a lead-time-based ATP rule or an infinite-availability-based ATP rule to determine promising results, allocation rules will not be considered.

If order promising is applying an ATP rule in supply chain availability search mode, an allocation rule may be applied if the following are true:

- The fulfillment line being promised has specified a demand class.
- The demand class has been specified in an allocation rule that is applicable to the item being promised.
- The ATP rule being applied has been defined to respect allocation constraints.

Sourcing Rules

When order promising is conducting a supply chain availability search, sourcing rules and the assignment sets that the sourcing rules are contained within provide the details of the supply chain to search. You may have many different manufacturing and distribution locations that can supply the same product. Sourcing rules determine the acceptable fulfillment locations to be considered. The order promising supply chain availability search results determine the best location, based on the product and order request date, from the locations specified in the sourcing rules.

Manage ATP Rules

ATP Rule Promising Modes: Explained

The promising mode of an available-to-promise (ATP) rule determines which set of attributes order promising logic evaluates when determining ATP results. When you create an ATP rule, the first thing that you must specify is which of the three promising modes is applicable to the ATP rule.

These are the three promising modes:

- Supply chain availability search
- Lead time based
- Infinite availability based

Infinite Availability Mode

You create ATP rules in infinite availability mode most often for assigning to items that are not constrained in supply and are of low value. Here are the key points of the infinite availability based mode:

- The item is promised on the requested date irrespective of availability.
- No availability search is performed.

- Calendars are respected during promising.
- The order promising engine does not generate any pegging.
- Transit time constraints are respected; therefore, requests within transit lead times are promised after accounting for the transit lead time.

No other attributes are associated with the infinite availability mode of promising.

Lead Time Mode

You create ATP rules in lead time promising mode most often for assigning to items that are always assembled or built on demand. This mode is typically used for items that can be procured or produced with a reliable lead time. Here are the key points of the lead time based mode:

- Orders are promised after a specified lead time, as promise dates are always offset from the requested date by the specified lead time.
- No supply availability search is performed.
- Lead time can be specified in multiple ways.
- The order promising engine does not generate any pegging.
- Calendars and transit time constraints are respected.

When you create ATP rules in lead time promising mode, you select one of the following four lead times to be used:

- Total lead time
- Cumulative manufacturing lead time
- Cumulative total lead time
- User-defined lead time

Supply Chain Availability Search

You create ATP rules in supply chain availability search mode for assigning to items for which you want promising results determined by a search for available product supply. The supply chain availability search promising mode provides complex and highly configurable promising functionality that enables you to configure how the search will be conducted. Here are the key points of the supply chain availability search mode:

- The search mode is a comprehensive and highly customizable mode of promising.
- Detailed availability search is performed across supply chain depending on the options that you select when you create the rule.
- Pegging information is generated by the engine.
- Lead times, calendars, capacities, transport modes, and supply chain network are considered during promising.

When you create ATP rules in supply chain availability search mode, you specify many additional attributes including these:

- Whether to search components and resources
- Whether to enable a profitable-to-promise search
- Whether to respect allocation constraints
- Which types of supply and demand to consider
- What lead time to use for the infinite availability fence
- How many days to consider for past-due demand and past-due supply

ATP Rule Precedence: How It Is Determined

You assign available-to-promise (ATP) rules to specific items, organizations, or item categories, or to combinations of specific items and organizations. When order promising is determining which ATP rule to use for the item being promised, there may be multiple rules that are applicable, but only one rule will be applied. For example, there may one ATP rule assigned to the item category of the item being promised, and another ATP rule assigned specifically to the item that is being promised. When there are multiple rules applicable to the item being promised, a hierarchical precedence is used to determine which rule will apply.

Settings That Affect ATP Rule Precedence

When assigning an ATP rule, you select an assignment basis for the assignment. The four choices for assignment basis are item category, item, organization, and item and organization combined.

How ATP Rule Precedence Is Calculated

When there are multiple ATP rules applicable to the item being promised, order promising uses a hierarchical precedence of least granular assignment basis to most granular assignment basis to determine which rule to apply. Order promising applies the rule with the most granular assignment basis. The following lists the four choices for assignment basis in order of least granular to most granular:

- Item category
- Item
- Organization
- Item and organization

Lead-Time-Based and Infinite-Availability-Based Promising Modes

Infinite-Availability-Based Promising Mode: How It Determines Promising Results

When applying an available-to-promise (ATP) rule in the infinite-availability-based promising mode, order promising bypasses supply consideration and determines the promise date from the requested date. Because supply is assumed to be infinite for all days, a request is always promised on the requested date, except in cases where transit lead times are violated.

Settings That Affect Infinite-Availability-Based ATP Mode

Which constraints order promising must respect when determining the promise date is determined by the type of requested date: requested arrival date or requested ship date.

How Infinite-Availability-Based Promising Results Are Determined

Order promising assumes infinite availability on all days, so no supply and demand matching is done for infinite promising. However, order promising must still respect certain constraints for the organization when determining the promise date from the requested date.

If the requested date is the requested arrival date, order promising must still respect calendar constraints and transit lead time constraints.

If the requested date is the requested ship date, order promising must still respect calendar constraints.

Lead-Time-Based Promising Mode: How It Determines Promising Results

In the lead-time-based promising mode, the promised date is delayed from the requested date based on the lead time defined in the available-to-promise (ATP) rule being applied for the item being promised.

Settings That Affect the Lead-Time-Based ATP Mode

The following two settings affect the promising results for lead-time-based promising:

- Lead time to be considered
- Type of requested date

The lead time to be considered is defined in the ATP rule being applied.

The type of requested date is defined in the fulfillment line being promised. The requested date is one of the following two types:

- Ship Date
- Arrival Date

How Lead-Time-Based Promising Results Are Determined

The calculations used to determine the promising results depend on the type of requested date:

- The requested date is the ship date.

Order promising applies the lead time offset defined in the ATP rule and schedules a promise on the requested date plus lead time offset after inflating the calendar date to account for any holidays. Order promising will not check whether there is availability of the item because even if there is availability of the item, the availability must be ignored.

- The requested date is the arrival date.

Order promising first derives the requested ship date by applying a transit time offset to determine the requested ship date. To determine the offset, the default carrier, shipping mode, and service level associated between the ship-from date and the ship-to date is considered. To derive the promised ship date, order promising then applies the lead time defined in the ATP rule to the derived requested ship date. Then to calculate the promised arrival date, order promising applies the transit time for the specified or default ship method. When determining dates, order promising accounts for any calendar constraints.

Promising Attributes for Lead Time Based Promising: Points to Consider

In the lead-time-based promising mode, the promised date will always be delayed from the requested date based on the lead time defined in the available-to-promise (ATP) rule being applied.

Lead Time Considered

When you create an ATP rule in the lead time based promising mode, you specify which one of the following four lead times is considered when the rule is applied:

- Total lead time
- Cumulative manufacturing lead time
- Cumulative total lead time
- User-defined lead time

Supply-Chain-Availability-Search-Based Promising Mode

Supply Chain Availability Search: How It Determines ATP

The goal of the supply chain availability search is always to find the available-to-promise (ATP) result that minimizes the lateness of the fulfillment line. The

options considered by specific supply chain availability search are determined by a number of settings. The results of the search are affected by which options are considered and by what factors must be evaluated when the options are considered.

Settings That Affect Supply Chain Availability Searches

The behavior of the supply availability search is primarily determined by the following four factors:

- Constraints specified on the fulfillment line, such as the specification of a ship-from warehouse and whether splits are allowed
- Attribute settings for the ATP rule that is being applied
- The supply chain defined by the assignment set in use and the sourcing rules that it contains
- Allocation constraints from an applicable allocation rule

Additional settings determine what additional options the supply chain availability search can consider. The following must be true for the supply chain availability search to consider capable-to-make when determining promising availability:

- The item is built from components, and the ATP rule has been enabled to search for components and resources.
- Inventory is maintained at the component level.
- Modeling of bills-of-material and routings have been collected into the order orchestration and planning data repository from the applicable fulfillment systems.

How Supply Chain Availability Searches Determine Promising Results

If the fulfillment line has many constraints specified, such as substitutions not allowed, the nature of the alternative options generated by the supply chain availability search changes. In the most constrained case, when a ship-from warehouse is specified and substitution and splits are not allowed, promising options are generated from only the specified ship-from warehouse for the specified item, possibly by considering different shipping methods that deliver the item to the customer site.

The fewer constraints specified on the fulfillment line, the more possibilities the supply chain availability search can consider. For example, if a ship-from warehouse is not specified, and splits and item substitutions are allowed, the supply chain availability search looks for the best possible ways of promising the fulfillment line by looking across all warehouses specified in the applicable sourcing rules and by considering splitting by date, or substituting items, or both. Order promising determines a default availability option as well as availability options that represent the best possible availability from each warehouse.

Unless the constraints on the fulfillment line restrict it from doing so, the supply chain availability search always considers the supply for the item at other

warehouses, also known as transfer capable-to-promise, and the supply for the item at suppliers, also known as buy capable-to-promise. If the ATP rule being applied has enabled the consideration of components and resources, the supply chain availability search considers the availability of the components and resources consumed during manufacturing, also known as make capable-to-promise. For example, if the settings enable a capable-to-promise search, and an end item is made of two components, C1 and C2, which are assembled on a resource R1, if supply is available for the components, but not for the end time, the fulfillment line is promised by using the available supply of the components and by considering the resource availability.

If the Profitable to Promise attribute is enabled for the ATP rule being applied, the supply chain availability search overrides sourcing priorities to respect the least-cost source that it can promise from. The costs considered when determining the most optimal location to source the promise from are the following:

- Standard cost at internal organizations
- Standard cost at supplier locations
- Cost of internal transfers between organizations
- Cost of transit from supplier to internal organizations
- Cost of transit from ship-from locations to customer sites by shipping method

In the case of make capable-to-promise, the following costs are also considered:

- The cost associated with resource consumption defined as cost per unit of resource consumed
- The cost of the components required to make the end item

Profitable to Promise Example

An item is being requested on a fulfillment line with the following quantity and date:

- Requested Quantity: 60 units
- Requested Date: 05-Feb-2011

Two warehouses, M1 and M2, both have 100 units of the requested item available, and the customer would receive the item on 05-Feb-2011 from both warehouses. M1 is the preferred warehouse per the applicable sourcing rule, but the ATP rule being applied has the Profitable to Promise attribute enabled. The availability search considers the cost of the item at each warehouse:

- M1 standard cost: \$20 per unit
- M2 standard cost: \$10 per unit

In this example, the ATP search overrides the warehouse with the higher priority in the sourcing rule. The promising result is 60 units from warehouse M2 to arrive at the client with no delay.

Promising Attributes for Supply Chain Availability Searches: Points to Consider

When creating an available-to-promise (ATP) rule in supply chain availability search mode, you define promising attributes to influence how order promising will determine fulfillment options and fulfillment option priority when applying the ATP rule to determine order promising results. You define whether order promising will search components and resources to find promising options that include making the item. You define whether order promising will include profitable to promise to determine which fulfillment option to use. You also define whether the ATP rule will override applicable allocation rules.

To define the promising attributes, you enable or disable the following attributes for each ATP rule that you create in the supply chain availability search mode:

- Search components and resources
- Enable profitable to promise search
- Respect allocation constraints

Search Components and Resources Attribute

Select the **Search components and resources** check box to enable order promising to consider whether the end item can be made using its component and resources if an item is not available at a requested location. When order promising is applying an ATP rule with search components and resources enabled, order promising can determine promising results by looking into the availability of the components and resources required to make the item being promised. In make-to-order environments, inventory is often not available for the ordered item, so it is necessary for order promising to look at components and resources to promise the order. Order promising respects all relevant calendars, lead times, and capacities when searching components and resources.

Enable Profitable to Promise Search Attribute

Select the **Enable profitable to promise search** check box to enable order promising to use the lowest total fulfillment cost option as the final factor for determining which fulfillment option to choose. Order promising always gives highest priority to demand satisfaction so that an order is never delayed for the sake of using a lower cost option. However, when order promising is applying an ATP rule with the profitable to promise search attribute enabled, if order promising finds more than one fulfillment option for the same fulfillment date, order promising determines the fulfillment cost, and sourcing priority may be overridden if the source with the lower cost has a lower priority. When determining cost, order promising considers the standard cost of an item at an internal organization or order promising considers the supplier costs, transfer costs, rolled up costs in the case of manufacturing, and delivery costs to the customer.

Respect Allocation Constraints Attribute

Select the **Respect allocation constraints** check box to direct order promising to apply allocation constraints defined by any allocation rules applicable to the item

being ordered. If you do not select the **Respect allocation constraints** check box, order promising will disregard allocation constraints when applying the ATP rule.

Defining an Infinite Availability Time Fence: Points to Consider

You define the infinite availability time fence to specify the horizon after which supply is considered to be infinite. Order promising considers any demand that falls beyond this time fence as available. When a promise is made at the infinite time fence, no real supplies are consumed.

Infinite Availability Time Fence

Order promising assumes infinite supply availability after the time period defined by the infinite availability fence. For requested dates after the time fence, order promising promises on the requested date without checking availability. If the requested date is beyond the infinite time fence, no real supplies are used for promising. For requested dates within the infinite availability time fence, order promising conducts a supply chain availability search.

You specify a user-defined number of days for a user-defined infinite availability time fence, or select from one following three lead times to define a lead time based infinite availability time fence:

- Total lead time
- Cumulative manufacturing lead time
- Cumulative total lead time

Tip

You should define an infinite availability time fence. If you do not define an infinite availability time fence, order promising uses a horizon of a year which incurs a large increase in the memory used by the order promising engine.

Defining Past-Due Supply and Demand Considerations: Points to Consider

You define past-due supply and past-due demand to limit the past-due supply and past-due demand considered when order promising determines promising results.

Past-Due Demand Considered

Past-due demand is a demand with a scheduled date earlier than the current date. Most past-due demands need to be considered and accounted for as they are expected to ship in the future. However, you may have a number of days of past-due beyond which you no longer consider the demand valid. You specify the number of days of past-due for past-due demand to be included when order promising determines promising results. Order promising does not consider any past-due demand due before the number of days you specify. If you do not

specify a value for past-due demand considered, all past-due demand will be considered.

Past-Due Supply Considered

Past-due supply is a supply, usually in the form of a purchase order, for which the expected date is earlier than the current date. Most past-due supplies need to be considered as the expectation of supply is still considered valid. You specify the number of days of past-due supply to be included when order promising determines promising results. Order promising does not consider any past-due supply expected before the number of days you specify. If you do not specify a value for past-due supply considered, all past-due supply will be considered.

Defining an ATP Time Fence: Points to Consider

To ensure that short term supply is preserved for orders within the short term, define an available-to-promise (ATP) time fence. By defining an ATP time fence, you prevent scenarios where longer term orders that come in first are pegged against existing supply, delaying the promise for orders that come in for the short term.

Requested Dates within the ATP Time Fence

Supply searches for the period within the ATP time fence look for existing supply before looking through the supply chain for item availability. In other words, the supply search looks for on-hand supply before looking for capable-to-promise supply.

Requested Dates outside the ATP Time Fence

Supply searches for dates outside of the ATP time fence look for capable-to-promise supply before looking for existing supply availability.

The ATP Time Fence

You specify a user-defined number of days for the ATP time fence, or select from one following three lead times to define a lead time based ATP time fence:

- Total lead time
- Cumulative manufacturing lead time
- Cumulative total lead time

Manage Planning Allocation Rules

Planning Allocation Rules: Explained

You use allocation rules to control how supply is allocated among various classes of demand. If allocation rules are not defined, order promising promises on a

first-come-first-serve basis. If orders for a lower priority customer come in first, the lower priority customers could consume scarce supply. If you have defined an allocation rule for a demand class, the allocation amount serves as the upper allocation constraint for that demand class for items the allocation rule has been assigned to.

You specify how supply is allocated using one of the following three specification types:

- Number: Fixed quantity is allocated to a demand class.
- Percentage: Percentage of total supply is allocated to a demand class.
- Ratio: Relative ratios between the various demand classes are used to divide up the total available supply within a week.

Controlling Allocation among Demand Classes

You determine how to allocate your supply among various classes of demand using one of the three specification types. For each fulfillment line with a demand class specified for which there is an applicable allocation rule, order promising applies the allocation rule when assessing the quantity requested against the quantity available. Orders are promised up to the allocation defined for the demand class.

Here is an example scenario.

- Allocation Rule: 20 percent to low priority customers, 80 percent to high priority customers.
- Supply: 100 on 19-Oct-2011, 100 on 26-Oct-2011

The following table is an example order with two fulfillment lines.

Fulfillment Line Number	Requested Item	Requested Date	Requested Quantity	Demand Class
1	WM101	19-Oct-2011	30	Low Priority Customer
2	WM101	19-Oct-2011	30	High Priority Customer

Results:

- Fulfillment Line 1: Delayed by 7 days since there is insufficient supply to allocate until additional supply is received on 26-Oct-2011.
- Fulfillment Line 2: No delay since there is sufficient supply for the requested date.

Planning Allocation Rule Precedence: How It Is Determined

You assign allocation rules to specific items, specific organizations, specific item categories, or combinations of specific items and organizations. When the

Order Promising process is determining whether an allocation rule applies to the item being promised, the process may determine that multiple rules are applicable. However, only one rule will be applied. For example, there may be one rule assigned to the item category of the item being promised, and another rule assigned to the item that is being promised. When there are multiple rules applicable to the item being promised, a hierarchical precedence is used to determine which rule will apply.

Settings That Affect Allocation Rule Precedence

When assigning an allocation rule, you select an assignment basis for the assignment. The four choices for assignment basis are item category, item, organization, and item and organization combined.

How Allocation Rule Precedence Is Determined

When there are multiple allocation rules applicable to the item being promised, order promising uses a hierarchical precedence of least granular assignment basis to most granular assignment basis to determine which rule to apply. Order promising applies the rule with the most granular assignment basis. The following lists the four choices for assignment basis in order of least granular to most granular:

- Item Category
- Item
- Organization
- Item and organization

Allocation Targets: Examples

Allocation targets can be defined using a specification type of number, percentage, or ratio. The following examples illustrate the use of each of these specification types.

Number Example

When you define allocation using the number specification type, you are specifying a fixed quantity to be allocated to the demand class. For example, if you want to allocate 500 units for a high priority demand class, you create an allocation rule using the number specification type with an allocation target of 500 for the high priority demand class.

Percentage Example

When you define allocation using the percentage specification type, you are specifying a percentage of overall item supply to be allocated to a demand class. For example, if you create an allocation rule using the percentage specification type with an allocation target of 20 for the low priority demand class, if total

supply is 1000, up to 200 units may be promised for the low priority demand class.

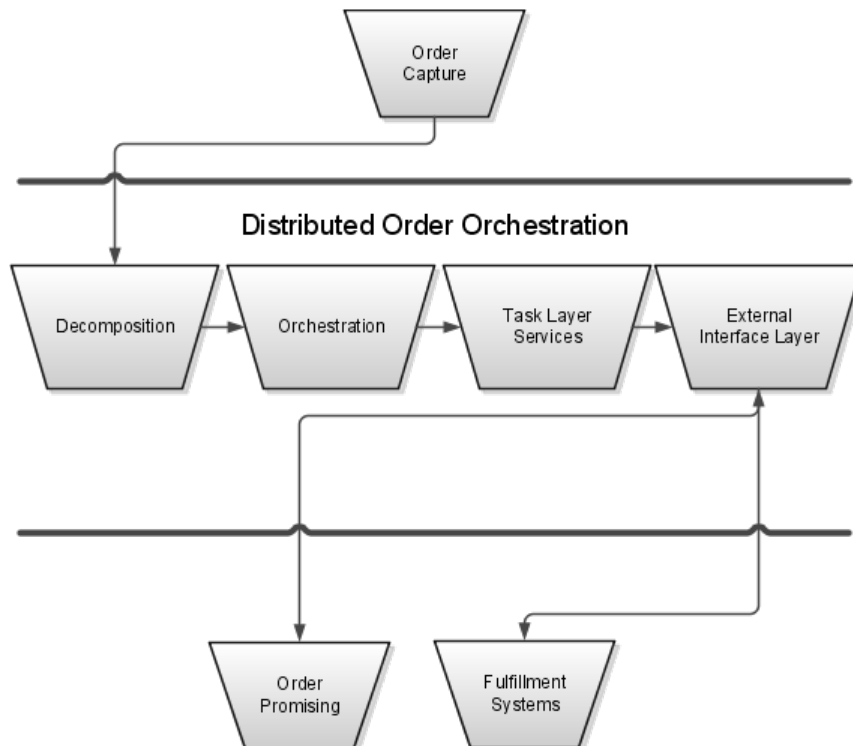
Ratio Example

When you define allocation using the ratio specification type, you are specifying that a relative ratio between the demand classes is used to divide up the total available supply within week. For example, if you create an allocation rule using the ratio specification type with a three to one ratio between high priority demands and low priority demands, and the total supply is 2000 units, the high priority demand class is allocated 1500 units, and the low priority demand class is allocated 500 units for the week.

Define Sales Order Fulfillment

Oracle Fusion Distributed Order Orchestration Components: How They Work Together

The Oracle Fusion Distributed Order Orchestration architecture is situated between one or more order capture systems and one or more fulfillment systems. When a sales order enters Distributed Order Orchestration, the application components process the order, first by breaking it down into logical pieces that can be fulfilled, then assigning an appropriate set of sequential steps to fulfill the order, and, finally, calling services to carry out the steps. Throughout the process, Distributed Order Orchestration continues to communicate with the order capture and fulfillment systems to process changes and update information. This figure shows the components that affect order processing. A sales order enters Distributed Order Orchestration from the order capture application. In Distributed Order Orchestration, the sales order proceeds through decomposition, orchestration, task layer services, and the external interface layer before proceeding to fulfillment systems. The following explanations fully describe the components within Distributed Order Orchestration.



Decomposition

During decomposition, the application breaks down the sales order and uses defined product transformation rules to transform the sales order into an orchestration order. Then the fulfillment lines are grouped and assigned to designated orchestration processes with step-by-step fulfillment plans. An orchestration process is a predefined business process that coordinates the orchestration of physical goods and activities within a single order and automates order orchestration across fulfillment systems.

Orchestration

Orchestration is the automated sequence of fulfillment steps for processing an order. The orchestration process provides the sequence and other important information, such as forward and backward planning, how to compensate for changes, and which statuses to use.

During orchestration, task layer services are called to carry out the steps of the orchestration process.

Task Layer Services

Task layer services execute user-defined fulfillment process steps and manage fulfillment tasks. These services send information to downstream fulfillment systems and interpret the responses and updates from those systems. For example, task layer service Create Shipment request is invoked by a ship order process to send a shipment request to the shipping system.

External Interface Layer

The external interface layer manages the communication between Distributed Order Orchestration and external fulfillment systems. Its primary functions are routing the fulfillment request and transforming the data.

Orchestration Lookups: Explained

Oracle Fusion Distributed Order Orchestration provides lookups that you can use to optionally define values during processing. The majority of lookups are system-level, and they cannot be changed. You can make certain changes to user-level and extensible lookups.

User-Level Lookups

Distributed Order Orchestration provides one user-level lookup: `DOO_ACTIVITY_TYPE`.

Users can:

- Insert new codes.
- Update the start date, end date, and enabled fields.
- Delete codes.
- Update tags.

Extensible Lookups

The following extensible lookups are provided:

- DOO_HLD_RELEASE_REASON
- DOO_MSG_REQUEST_FUNCTION
- DOO_RETURN_REASON
- DOO_SUBSTITUTION_REASON

With extensible lookups, users can:

- Insert new codes.
- Update the start date, end date, enabled fields and tag, but only if the code is not seed data.
- Delete codes, but only if the code is not seed data.

Users cannot:

- Update the module.
- Delete the lookup type.

Orchestration Profile Management: Points to Consider

Oracle Fusion Distributed Order Orchestration provides several product-specific profile values. Some control behavior in the Order Orchestration work area, while others control the receipt and transformation of sales orders into orchestration orders. Most have predefined values, so you do not need to configure them, unless your organization requires different profile values.

Currency Conversion Type

This profile option defines the value to use during any currency conversion in the Order Orchestration work area. The value is a conversion type. You can update the profile option at the site and user levels.

Display Currency

This profile option defines the currency used to display the amount in the Order Orchestration work area. The value is a currency. You can update the profile option at the site and user levels.

Required Overview Status Filter

This profile option defines the default customer used to filter the summary of status data on the Overview page of the Order Orchestration work area and allows the user to view summary data only one customer at a time by removing the All option. No value is provided, by default. If you need to use it for performance reasons, then enter one of your customer IDs. You can update the profile option at the site level.

Retain Sales Order Number for Orchestration Order Number

This profile option specifies whether to use the sales order number as the orchestration order number during sales order transformation. The default value is N. You can update the profile at the site and user levels.

User Request Waiting Period in Seconds

This profile option specifies the number of seconds to wait after an action is taken to allow asynchronous services to complete before presenting a confirmation or warning message in the Order Orchestration work area. The default value is 5. You can update the profile option at the site level.

Oracle Fusion Distributed Order Orchestration Extensible Flexfields: Explained

An extensible flexfield is similar to a descriptive flexfield in that it is an expandable data field that is divided into segments, where each segment is represented in the application database as a single column. Extensible flexfields support a one-to-many relationship between the entity and its extended attribute rows. Using extensible flexfields, you can add as many context-sensitive segments to a flexfield as you need. You can set up extensible flexfields for a fulfillment line or on other entities that support extensible flexfields. Extensible flexfields are useful primarily when you need to segregate attributes by task layer or capture multiple contexts to group them based on function.

Transactional Entities That Support Extensible Flexfields

You can use extensible flexfields for the following transactional entities on the orchestration order object.

- Headers
- Orchestration order lines
- Fulfillment lines
- Fulfillment line details
- Price adjustments
- Sales credits
- Payments
- Lot serial
- Activities

Oracle Fusion Distributed Order Orchestration Extensible Flexfield Uses: Explained

Use extensible flexfields to send and receive additional information between Oracle Fusion Distributed Order Orchestration and integrated applications, write

business rules, process changes, and display additional attributes on the Order Orchestration work area.

Receive Additional Information or Attributes from an Order Capture Application

The sales order that Oracle Fusion Distributed Order Orchestration receives contains a predefined set of attributes. Your business process may require that you capture additional information or attributes on the sales order to use during order fulfillment. Distributed Order Orchestration uses extensible flexfields to receive the additional set of information or attributes that are captured on the sales order and use them during the fulfillment orchestration process.

Send Additional Information Relevant to a Fulfillment Execution Application

The task layers use a specific fulfillment request object to initiate a fulfillment request in a downstream application. Using extensible flexfields, Distributed Order Orchestration can pass any additional information that you set up and capture on the orchestration order, beyond the predefined set of attributes, during implementation.

Receive Additional Information from a Fulfillment Execution Application

During the response to a fulfillment request, a fulfillment execution application can send various attributes that may have business value and which need to be seen either from the Order Orchestration work area or in the order capture application. This additional information also can be used in the next set of tasks, if that information is relevant to the set of tasks that follow this task. Using extensible flexfields, Distributed Order Orchestration can receive additional sets of attributes from the fulfillment execution applications.

Write Business Rules

You can use extensible flexfield attributes to write business rules for Distributed Order Orchestration. You can use extensible flexfield attributes to write rules for the following rules implementations:

- Transformation rules
- Pretransformation and posttransformation defaulting rules
- Process assignment rules
- External interface routing rules

Process Changes

You can use extensible flexfields during change management. You can designate an extensible flexfield as an order attribute that indicates that a change occurred. An extensible flexfield is interpreted as a single unit for change processing. Changes are not allowed from the Order Orchestration work area and are supported only through the services.

Display Attributes in the Order Orchestration Work Area

The Order Orchestration work area displays the following extensible flexfields.

- Headers
- Orchestration order lines
- Fulfillment lines

- Fulfillment line details
- Activities

The extensible flexfield attributes are read-only. Users cannot edit them in the Order Orchestration work area.

Oracle Fusion Distributed Order Orchestration Extensible Flexfield Setup: Explained

To set up Oracle Fusion Distributed Order Orchestration extensible flexfields, you must define flexfields, deploy them, synchronize them with business rules, synchronize the SOA artifacts, and configure the enterprise business object.

The specific steps follow:

1. Run the Publish Extensible Flexfields Attributes process to create categories for the extensible flexfields.
2. Define categories, contexts, and associated segments along with value sets for each extensible flexfield that you want to enable through the Manage Extensible Flexfields setup.
3. Deploy the flexfield.
4. Run the Publish Extensible Flexfields Attributes process to synchronize the extensible flexfield attributes with Oracle Business Rules.
5. Execute the SOA composite UpdateSOAMDS for synchronizing the SOA artifacts.
6. Extend the enterprise business object.
7. Map the enterprise business object attributes with the extensible flexfield attributes.

Extending the Enterprise Business Object

Oracle Fusion Distributed Order Orchestration uses enterprise business objects to interact with external systems. An enterprise business object is made up of a business component, shared components, common components, reference components, common enterprise business objects, choice components, and attributes. These components are nested as required to create a sophisticated content model with varying cardinality from zero to one or unbounded. A custom element is defined in these component types and can be used to extend the properties of the component. The custom element then can be further mapped to extensible flexfield attributes in the interfaces.

The enterprise business objects are delivered as a set of XSD files. For every enterprise business object, a custom XSD file is provided in which all customer extensions are stored. Using customer extensions, you can include on the sales order additional attributes that your organization needs. For example, assume that you want to add DeliverToParty to the order header because the shipping system can honor this information. To integrate with the shipping system, you must extend the Sales Order enterprise business object. To add this new attribute at the header level, edit the following part of the CustomSalesOrderEBO.xsd schema definition:


```
<xsd:complexType name="CustomSalesOrderScheduleType"/>
```

After adding the attributes, this section of the schema definition looks like:

```
<xs:complexType name="CustomSalesOrderScheduleType">  
<xs:sequence>  
<xsd:element ref="corecom:DeliverToPartyReference" minOccurs="0">  
</xsd:element>  
</xs:sequence>  
</xs:complexType>
```

The Sales Order enterprise business object is now ready to carry the custom attributes for DeliverToPartyReference. The custom attributes can be either from the common components library or can be new elements or attributes that are directly added if they did not exist in the common components library. Note that the extension of the underlying Sales Order enterprise business object also extends all enterprise business messages that reference the Sales Order enterprise business object. In the case of the Receive and Transform service that is used by the order capture application to submit an order, this is ProcessSalesOrderFulfillment.

Transformation Extensions

The default transformations for the existing schemas may not be sufficient for some of your organization's specific business operations. You might want to add elements to the enterprise business object schemas as explained previously and then change transformation maps for the newly added elements to transfer the information from one application to the other.

At implementation time, the transformation maps that are associated with the external-facing interfaces must be modified to map the extensible flexfield attributes to the enterprise business object attributes.

Manage Hold Codes

Hold Definitions: Explained

Holds pause action on the business objects and services to which they are applied. In Oracle Fusion Distributed Order Orchestration, holds can come from an order capture system or the Order Orchestration work area. You define codes for use throughout Distributed Order Orchestration. The codes you define in Distributed Order Orchestration are for holds that originate in this application only. When you define hold codes, you indicate to which services the hold can be applied. You can also create a hold code that applies a hold to all services. Task layer services check to see whether any hold code is attached to the fulfillment line or order for one or more tasks in the orchestration process.

A hold that is applied in Distributed Order Orchestration can be released by the same application only, either by a user or by an orchestration process. A hold is applied by an orchestration process only when there is an existing hold request, either from the order capture application or from the Order Orchestration work area user. For example, an orchestration process is at the scheduling step when an order capture user sends a request to hold the shipping task. Distributed Order Orchestration stores the request until the orchestration process gets to the shipping step. At that point, the application searches for existing requests and applies them. When an orchestration process is canceled, associated holds are

released automatically. Otherwise, the Order Orchestration user must release holds manually.

Only an order capture user can release a hold applied in the order capture application.

Hold Codes from Other Applications

When a hold enters Distributed Order Orchestration from an order capture or order fulfillment application, it is transformed and becomes part of the orchestration order.

Manage Orchestration Source Systems

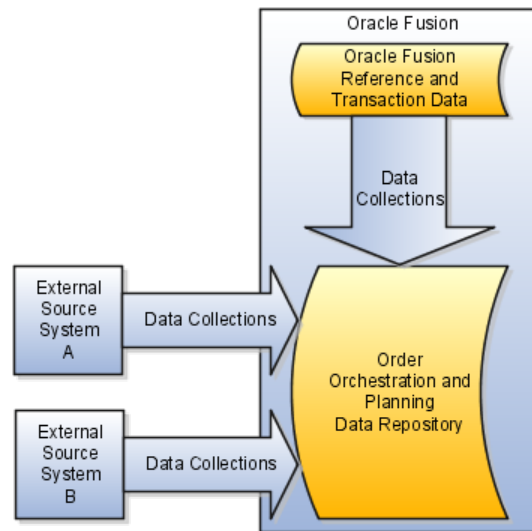
Managing Data Collection Source Systems: Explained

To populate the order orchestration and planning data repository, you collect data from external source systems, such as external fulfillment source systems and external order capture source systems, and from the Oracle Fusion source system. You manage which source systems are data collection source systems by defining collections parameters and enabling which source systems allow collections.

You manage two categories of source systems for data collections:

- External source systems
- The Oracle Fusion source system

The following figure illustrates data collections from three source systems. Two of the source systems are external source systems. One of the source systems is the Oracle Fusion source system.



External Source Systems

Your business may have many external fulfillment and external order capture source systems. For each external source system from which you need to collect

data to include in the order orchestration and planning data repository, define the data collection parameters, and enable the source system for collections. For the Version data collection parameter, the choices are Other or Oracle Fusion.

The Oracle Fusion Source System

The order orchestration and order promising processes use data stored in the order orchestration and planning data repository. Some of the data that needs to be in the repository originates in the Oracle Fusion source system. To collect data from the Oracle Fusion source system, include the Oracle Fusion source system as a source system for data collection. Define the data collection parameters for the Oracle Fusion source system, and enable the source system for collections.

Defining Data Collection Parameters: Points to Consider

For each system from which you intend to collect data to populate the order orchestration and planning data repository, you define and maintain the source system data collection parameters.

For each source system, you complete the following for the data collection parameters:

- Specify the time zone.
- Specify the version, order orchestration type, and planning type.
- Define the number of database connections, parallel workers, rows per processing batch, and cached data entries.
- Enable collections allowed.
- Enable data cross-referencing.

Specify the Time Zone

You must specify the time zone for the source system because the time stamps contained in collected data are converted from the time zone used in the source system to the time zone used for all data stored in the order orchestration and planning data repository. Using the same time zone for all data stored in the order orchestration and planning data repository facilitates correct results when calculations are performed using attributes that store dates. For example, if the source system uses the US Eastern time zone, but the order orchestration and planning data repository stores all data in the US Pacific time zone, then a supply with a due date and time of July 10th 04:00 PM in the source system is stored in the order orchestration and planning data repository with a due date of July 10th 01:00 PM.

Specify the Version, Order Orchestration Type, and Planning Type

You must define one, and only one, source system with the Version attribute equal to Oracle Fusion and the Order Orchestration Type attribute equal to Order Orchestration.

You may define many source systems with the Version attribute equal to Other. For the source systems with the Version attribute equal to Other, the Order Orchestration Type attribute can equal Fulfillment or Order Capture and the Planning Type attribute can equal Fulfillment. Any combination of these values is allowed to describe the purpose of the source system, but you must provide a value for at least one of these type parameters. These parameters do not impact the behavior of the collections process.

Note

Once you have saved a system with the Version attribute equal to Oracle Fusion, you cannot change the value for the Version attribute.

Note

You cannot change the version of a source system from Others to Fusion. You must delete the planning source system definition by scheduling the Delete Source Configuration and All Related Data process. The Delete Source Configuration and All Related Data process performs multiple steps. First the process deletes all data previously collected from the source system. After deleting the collected data, the process deletes the planning source system definition and collection parameters. After the Delete Source Configuration and All Related Data process completes, you must redefine the planning source system definition on the Manage Planning Source Systems page.

Define the Number of Database Connections, Parallel Workers, Rows Per Processing Batch, and Cached Data Entries

These parameters affect the usage of system resources. The table below defines what each parameter does and provides guidelines for setting it.

Parameter	What the Parameter Does	A Typical Value for the Parameter
Number of Database Connections	Defines the maximum number of database connections the source server can create during the collection process. This controls the throughput of data being extracted into the Source Java program.	10
Number of Parallel Workers	Defines the maximum number of parallel workers (Java threads) used to process the extracted data. The number here directly impacts the amount of CPU and memory used during a collection cycle.	30
Number of Rows per Processing Batch	Define the number of records to process at a time. The idea is to allow the framework to process data in byte-size chunks. A batch too small may cause extra overhead while a batch too big might peak out memory or network bandwidth.	10,000

Cached Data Entries in Thousands	During data collections, various lookup and auxiliary data are cached in the collection server to support validation. For example, currency rate may be cached in memory. This parameter controls the maximum number of lookup entries cached per lookup to prevent the server from occupying too much memory.	10,000
---	--	--------

Enable Collections Allowed

Before enabling a source system for collections, ensure your definition of the other parameters are complete for the source system. Ensure you have defined values for all applicable attributes, and where applicable, you have enabled organizations for collections or for ATP Web services.

Enable Data Cross-Referencing

When you enable a source system for data cross-reference, the data collections from the source system perform additional processing steps to check for and to cross-reference data during data collections. You must enable cross-referencing for Order Capture source systems.

Enabling Organizations for Data Collections: Points to Consider

From the list of organizations for each source systems, you designate which organizations will have their data collected when a collections process collects data from the source system.

Deciding Which Organizations to Enable for Collections

To determine which organizations to enable for collections, analyze the sourcing strategies for your company, the type of organization for each organization in the list, and any other business requirements that would determine whether system resources should be expended to collect data from that organization. If the data from that organization would never be used by order promising or order orchestration, no need to collect the data.

For example, consider a scenario where the list of organizations for a source system includes 20 manufacturing plants and 10 distribution centers. Because the business requirements specify that the movements of materials from the manufacturing plants to the distribution centers are to be controlled separately from order orchestration and order promising, there are no sourcing rules that include transferring from one of the manufacturing plants. For this scenario, you would only enable the 10 distribution centers for collections.

FAQs for Orchestration Source Systems

Can I add a new source system to the list of data collection source systems?

No. You cannot add additional source systems when managing source systems for data collections for the order orchestration and planning data repository.

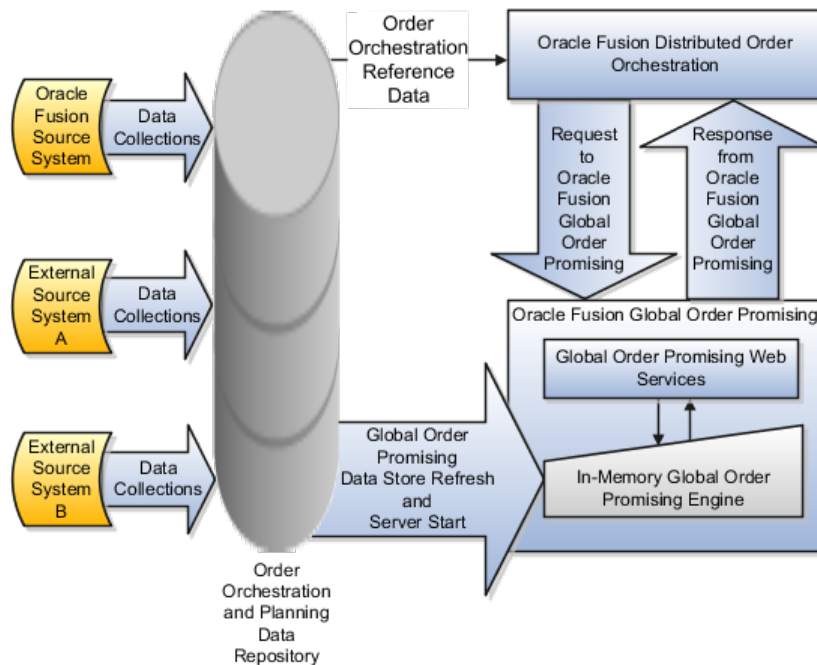
Source systems must first be defined in the Trading Community Model. For the system to be listed as one of the systems from which to choose from when managing source systems, the definition of the system in the Trading Community Model must enable the system for order orchestration and planning.

Collect Orchestration Reference and Transaction Data

Data Collections, Order Orchestration, and Order Promising: How They Fit Together

You perform data collections to populate the order orchestration and planning data repository. The collected data is used by Oracle Fusion Distributed Order Orchestration and Oracle Fusion Global Order Promising.

The following figure illustrates that the order orchestration and planning data repository is populated with data from external source systems and from the Oracle Fusion source system when you perform data collections. Oracle Fusion Distributed Order Orchestration uses some reference data directly from the repository, but the Global Order Promising engine uses an in-memory copy of the data. After data collections are performed, you refresh the Global Order Promising data store with the most current data from the data repository and start the Global Order Promising server to load the data into main memory for the Global Order Promising engine to use. When Oracle Fusion Distributed Order Orchestration sends a scheduling request or a check availability request to Oracle Fusion Global Order Promising, the Global Order Promising engine uses the data stored in main memory to determine the response.



Data Collections

You perform data collections to populate the order orchestration and planning data repository with data from external source systems and from the Oracle Fusion source system.

Order Orchestration

Oracle Fusion Distributed Order Orchestration uses some reference data directly from the order orchestration and planning data repository. You must perform data collections for the order orchestration reference entities even if you are not using Oracle Fusion Global Order Promising.

Important

Before collecting data from an Oracle Fusion source system, you must define at least one organization for the source system. After you have defined at least one organization for the source system, you must update the organization list for the source system on the Manage Planning Source Systems page or Manage Orchestration Source Systems page, and enable at least one organization for collections. If there are no organizations enabled for collections when a collections process runs, the collections process will end with an error.

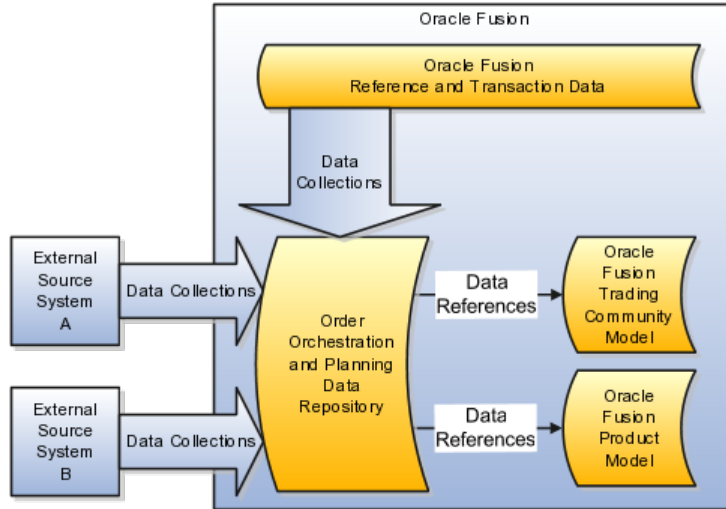
Order Promising

The Global Order Promising engine uses an in-memory copy of the data from the order orchestration and planning data repository. When Oracle Fusion Distributed Order Orchestration sends a scheduling request or a check availability request to Oracle Fusion Global Order Promising, the Global Order Promising engine uses the data stored in main memory to determine the response to send back to order orchestration. After a cycle of data collections is performed, you refresh the Global Order Promising data store with the most current data from the data repository and start the Global Order Promising server to load the data into main memory for the Global Order Promising engine to use.

Collecting Data for the Order Orchestration and Planning Data Repository: Explained

The order orchestration and planning data repository provides a unified view of the data needed for order orchestration and order promising. You manage data collection processes to populate the data repository with data collected from external source systems and from the Oracle Fusion source system. You manage the data collection processes to collect the more dynamic, transaction data every few minutes and the more static, reference data on a daily, weekly, or even monthly schedule. The data collected into the data repository contains references to customer data managed in the Oracle Fusion Trading Community Model and to product data managed in the Oracle Fusion Product Model. The data managed in these models is not collected into the order orchestration and planning data repository.

The following figure illustrates that the order orchestration and planning data repository is populated with data collected from external source systems and from the Oracle Fusion source system. The data repository does not contain data managed by the Oracle Fusion Trading Community Model and the Oracle Fusion Product Model. The data collected into the data repository references data managed in these models.



When you plan and implement your data collections, you determine:

- Which entities you collect from which source systems
- The frequency of your collections from each source system
- Which data collection methods you will use to collect which entities from which source systems
- The sequences of your collections

Consider these categories of data when you plan your data collections:

- Data collected for order promising
- Data collected for order orchestration
- Data not collected into the order orchestration and planning data repository

Data Collected for Order Promising

The following categories of data are collected and stored to support order promising:

- Existing supply including on-hand, purchase orders, and work orders
- Capacity including supplier capacity and resource capacity
- Related demands including work order demands and work order resource requirements
- Planned supply including planned buy and make orders
- Reference data including calendars, transit times, and routings

For a specific collection, the data that is collected is determined by the data collection entities specified for that collection.

Important

After performing data collections, you must refresh the Order Promising data store and start the Order Promising engine to ensure the engine is using the data most recently collected.

Data Collected for Order Orchestration

The following reference data is collected and stored to support order orchestration:

- Warehouses (Organization Parameters)
- Units of Measure
- Carriers
- Currencies
- Currency Conversions
- Shipping Methods
- Order Orchestration Reference Objects
 - Payment Term
 - Accounting Rule
 - Invoicing Rule
 - Sales Credit Type
 - AR Receipt Method
 - Document Category
 - Service Level
 - Mode of Transport
 - Tax Classification Code
 - Return Reason
 - Shipment Priority
 - Payment Method
 - Freight Terms
 - Tax Exemption Reasons
 - Activity Types
 - FOB

Tip

Use the Review Planning Collected Data page or the Review Order Orchestration Collected Data page to explore many of the entities and attributes collected for the order orchestration and planning data repository.

Data Not Collected into the Order Orchestration and Planning Data Repository

Data collected into the order orchestration and planning data repository includes attributes, such as customer codes, that refer to data not collected into the

data repository. Most of the data references are to data in the Oracle Fusion Trading Community Model or in the Oracle Fusion Product Model. Some of the data references are to data outside the models, such as item organizations and inventory organizations. To manage data collections effectively, especially the sequences of your collections, you must consider the data dependencies created by references to data not collected into the data repository.

References to data in the Oracle Fusion Trading Community Model include references to the following:

- Source systems
- Geographies and zones
- Customers
- Customer sites

References to data in the Oracle Fusion Product Model include references to the following:

- Items, item relationships, and item categories
- Item organization assignments
- Structures

Data Collection Entities: Explained

When you collect data for the order orchestration and planning data repository, you specify which of the data collection entities to collect data for during each collection. When you plan your data collections, you plan which entities to collect from which source systems and how frequently to collect which entities. One of the factors you include in your planning considerations is the categorizations of each entity. One way entities are categorized is as reference entities or transaction entities. You typically collect transaction entities much more frequently than reference entities.

Another way entities are categorized is as source-specific entities or global entities. For global entities the order in which you collect from your source systems must be planned because the values collected from the last source system are the values that are stored in the data repository.

When you plan your data collections, you consider the following categorizations:

- Source-specific entities
- Global entities
- Reference entities
- Transaction entities

You also consider which entities can be collected from which types of source systems using which data collection methods as follows:

- Entities you can collect from the Oracle Fusion source system and from external source systems
- Entities you can collect only from external source systems

Source-Specific Entities

When you collect data for a source-specific entity, every record from every source system is stored in the order orchestration and planning data repository. The source system association is maintained during collections. The data stored in the data repository includes the source system from which the data was collected.

For example, you collect suppliers from source system A and source system B. Both source systems contain a record for the supplier named Hometown Supplies. Two different supplier records will be stored in the data repository for the supplier named Hometown Supplies. One record will be the Hometown Supplies supplier record associated with source system A and the second record will be the Hometown Supplies supplier record associated with source system B.

The majority of the data collections entities are source-specific entities.

Global Entities

When you collect data for a global entity, only one record for each instance of the global entity is stored in the order orchestration and planning data repository. Unlike source-specific entities, the source system association is not maintained during collections for global entities. The data stored in the data repository for global entities does not include the source system from which the data was collected. If the same instance of a global entity is collected from more than one source system, the data repository stores the values from the last collection.

For example, you collect units of measure (UOM) from three source systems and the following occurs:

1. During the collection of UOM from source system A, the Kilogram UOM is collected.

This is first time the Kilogram UOM is collected. The Kilogram record is created in the data repository.

2. During the collection of UOMs from source system B, there is no collected UOM with the value = Kilogram

Since there was no record for the Kilogram UOM in source system B, the Kilogram record is not changed.

3. During the collection of UOMs from source system C, the Kilogram UOM is also collected.

Since the collections from source system C include the Kilogram UOM, the Kilogram record in the data repository is updated to match the values from source system C.

The following entities are the global entities:

- Order orchestration reference objects
- Units of measure (UOM) and UOM conversions
- Demand classes
- Currency and currency conversion classes

- Shipping methods

Tip

When you collect data for global entities from multiple source systems, you must consider that the last record collected for each occurrence of a global entity is the record stored in the order orchestration and planning data repository. Plan which source system you want to be the source system to determine the value for each global entity. The source system that you want to be the one to determine the value must be the source system that you collect from last.

Reference Entities

Reference entities are entities that define codes and valid values that are then used regularly by other entities. Units of measure and demand classes are two examples of reference entities. Reference entities are typically static entities with infrequent changes or additions. Whether an entity is reference entity or a transaction entity does not impact how it is stored in the order orchestration and planning data repository.

You consider whether an entity is a reference entity or a transaction entity when determining which collection method to use to collect data for the entity. You typically use the staging tables upload method to collect data for reference entities from external source systems. You typically used the targeted collection method to collect data for reference entities from the Oracle Fusion source system unless the reference entity is one of the entities for which the targeted collection method is not possible.

Transaction Entities

Transaction entities are the entities in the data repository that store demand and supply data. Because the data for transaction entities changes frequently, you typically use the web services upload method to collect data for transaction entities from external source systems. You typically use the continuous collection method to collect data for transaction entities from the Oracle Fusion source system.

Entities You Can Collect From the Oracle Fusion Source System and From External Source Systems

Many of the data collection entities can be collected from both types of sources systems. For the following entities you can use any of the collections methods:

- Approved supplier lists
- Calendars
- Calendar associations
- Interlocation shipping networks
- Item costs
- On hand
- Organization parameters
- Purchase orders and requisitions

- Subinventories
- Suppliers
- Units of measure

For the following entities you can only use the Web service upload method to collect data from external source systems:

- Currencies
- Order orchestration reference objects
- Shipping methods

Entities You Can Collect only From External Source Systems

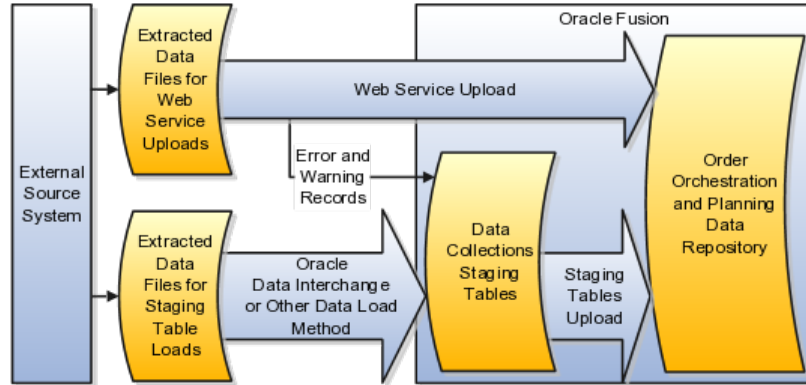
Many of the data collection entities can be only collected from external sources systems. For these entities, you can use both methods for collecting data from external source systems. Remember to consider frequency of change and volume of data in your considerations of which methods to use to collect which entities. The following are the entities you can only collect from external sources systems:

- Customer item relationships
- Demand classes
- Planned order supplies
- Routings
- Resources
- Resource availability
- Sourcing
- Supplier capacities
- Work-in-process supplies
- Work-in-process component demands
- Work-in-process resource requirements

Data Collection Methods for External Source Systems: Explained

To populate the order orchestration and planning data repository with data collected from external source systems, you use a combination of two data collection methods. The two methods are Web service uploads and staging tables uploads.

The following figure illustrates the two data collection methods, Web service uploads and staging tables uploads, used to collect data from external source systems. The figure illustrates that both methods require programs to be written to extract data from the external source systems. For Web service uploads, you load the data from the extracted data files directly into the order orchestration and planning data repository. Any records with errors or warnings are written to the data collections staging tables. For staging table uploads, you load the data from the extracted data files into the data collections staging tables, and then you use the Staging Tables Upload program to load the data from the staging tables into the data repository.



You determine which entities you collect from which source systems and at what frequency you need to collect the data for each entity. The data for different entities can be collected at different frequencies. For example, supplies and demands change frequently, so collect data for them frequently. Routings and resources, are more static, so collect data for them less frequently.

Which data collection method you use for which entity depends upon the frequency of data changes as follows:

- Web service upload
 - Use for entities with frequent data changes.
- Staging tables upload
 - Use for entities with more static data.

Web Service Upload Method

Use the Web service upload method for entities that change frequently, such as supply and demand entities. You determine the frequency of collections for each entity. For certain entities, you may implement Web services to run every few minutes. For other entities, you may implement Web services to run hourly.

To implement and manage your Web service uploads, you must design and develop the processes and procedures to extract the data in the format needed by the data collection web services. For more information regarding the data collection Web services, refer to the Oracle Enterprise Repository. For additional technical details, see Oracle Fusion Order Promising Data Collection Staging Tables and Web Service Reference, document ID 1362065.1, on My Oracle Support at <https://support.oracle.com>.

Staging Tables Upload Method

Use the staging tables upload method for entities that do not change frequently, such as routings and resources. You determine the frequency of collections for each entity. You may establish staging table upload procedures to run daily for some entities, weekly for some entities, and monthly for other entities.

To implement and manage your staging table uploads, you must develop the processes and procedures you use to extract data from an external source system. You use Oracle Data Interchange, or another data load method, to load the extracted data into the data collection staging tables. For additional technical details, such as the table and column descriptions for the data collection staging

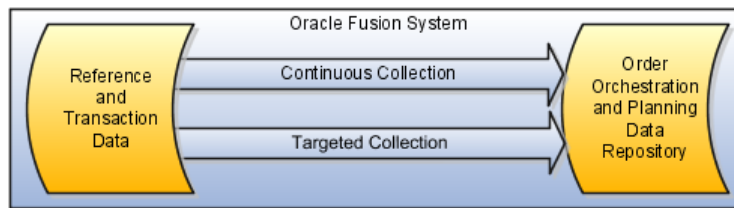
tables, see Oracle Fusion Order Promising Data Collection Staging Tables and Web Service Reference, document ID 1362065.1, on My Oracle Support at <https://support.oracle.com>.

For the final step of the staging tables upload method, you initiate the Load Data from Staging Tables process from the Manage Data Collection Processes page or via the Enterprise Scheduling Service.

Data Collection Methods for the Oracle Fusion Source System: Explained

To populate the order orchestration and planning data repository with data collected from the Oracle Fusion source system, you use a combination of two data collection methods: continuous collection and targeted collection. You typically use continuous collection for entities that change frequently and targeted collection for entities that are more static.

The following figure illustrates the two data collection methods, continuous collection and targeted collection, used in combination to collect data from the Oracle Fusion source system.



Continuous Collection

When you use the continuous collection method, you are only collecting incremental changes, and only for the entities you have included for continuous collection. Because continuous collection only collects incremental changes, you usually set up the continuous collection to run frequently, such as every five minutes.

Note

Prior to including an entity for continuous collection, you must have run at least one targeted collection for that entity.

Targeted Collection

When you collect data using the targeted collection method, you specify which entities to include in the targeted collection. For the included entities, the data in the data repository that was previously collected from the Oracle Fusion source system is deleted and replaced with the newly collected data. The data for the entities not included in the targeted collection is unchanged. You typically use the targeted collection method to collect data from entities that do not change frequently.

Manage Orchestration Data Collection Processes

Managing Data Collection Processes: Overview

For your data collections from the Oracle Fusion source system, you use the Manage Planning Data Collection Processes page or the Manage Orchestration Data Collection Processes page. From these pages you perform the following:

- Manage your continuous collections from the Oracle Fusion source system.
- Manage your collections destination server.
- Perform your targeted collections from the Oracle Fusion source system.

For your data collections from external source systems, most of the management of your Web services uploads and staging tables uploads is performed external to the Oracle Fusion application pages. If you choose to perform staging tables uploads, you initiate the Perform Data Load process from the Manage Planning Data Collection Processes page, from the Manage Orchestration Data Collection Processes page, or from the Oracle Fusion Enterprise Scheduler.

Continuous Collection Publish Process: Explained

To enable continuous collections, you must set up the publish data processes for the Oracle Fusion source system. The publish process performs the incremental data collections from the Oracle Fusion source system. You can start, stop, and pause the publish process. To review statistics regarding the publish process, view process statistics from the **Actions** menu on the Continuous Collection - Publish tab on the Manage Planning Data Collection Processes page or the Manage Orchestration Data Collection Processes page.

Note

Because continuous collections only collect net changes, you must perform at least one targeted collection for an entity before you include the entity for continuous collections.

Publish Process Parameters: Points to Consider

You define the publish process parameters to determine the frequency and scope of the continuous collections publish process.

You define the frequency and scope of continuous collections by specifying the following:

- Process Parameters
- Process Entities

Process Parameters

You determine how frequently the continuous collections publish process executes by specifying the frequency in minutes. The continuous collections publish process will publish incremental changes based on the frequency that was defined when the publish process was last started.

You determine which organizations will be included in the set of organizations for which data is collected by specifying an organization collection group. You can leave it blank if you want data collected from all organizations.

Process Entities

You determine which entities are collected during the continuous collections cycles by selecting which entities you want included in the collections. The continuous collections publish process collects incremental changes for the business entities that were included when the publish process was last started.

Collections Destination Server: Explained

The collections destination server is applicable to all four data collection methods. For the continuous collections method the collections server is the subscriber to the continuous collections publish process. From the Actions menu on the Collections Destination Server tab you can access a daily statistic report with statistics regarding each of the collection methods. You also can access a data collections summary report.

Destination Server Collections Parameters: Points to Consider

The collection parameters are initially set to what was defined for the Oracle Fusion system when your planning source systems or order orchestration source systems were initially managed. You can fine tune the parameters for your data collections.

Data Collection Parameters

The data collection parameters affect the usage of system resources. This table define what each parameter does and provides guidelines for setting it.

Parameter	What the Parameter Does	A Typical Value for the Parameter
Number of Database Connections	Defines the maximum number of database connections the source server can create during the collection process. This controls the throughput of data being extracted into the Source Java program.	10
Number of Parallel Workers	Defines the maximum number of parallel workers (Java threads) used to process the extracted data. The number here directly impacts the amount of central processing units and memory used during a collection cycle.	30
Cached Data Entries in Thousands	During data collections, various lookup and auxiliary data are cached in the collection server to support validation. For example, currency rate may be cached in memory. This parameter controls the maximum number of lookup entries cached per lookup to prevent the server from occupying too much memory.	10,000

Cross-Referencing Data During Data Collections: Explained

When you collect data from multiple source systems, you often collect a variety of values for the same instance of an entity. You cross-reference data during data collections to store a single, agreed value in the order orchestration and planning data repository for each instance of a global entity.

The following information explains why you might need to cross-reference your data during data collections, and what you need to do to implement cross-referencing:

- Cross-reference example
- Cross-reference implementation

Cross-Reference Example

The following table provides an example of why you might need to cross-reference your data during data collections. In the example, the Kilogram unit of measure is collected from two source systems. The source systems use a different value to represent kilogram. You decide to store kg for the value for Kilogram in the order orchestration and planning repository.

Source System	Collections Entity	Source Value	Target Value
System A	Unit of measure	kilogram	kg
System B	Unit of measure	k.g.	kg

Cross-Reference Implementation

To implement cross-referencing, you must complete the following actions:

1. Decide which business object to enable for cross-referencing.
2. For each object, work with business analyst to decide which values to map to which other values.
3. Update the domain value maps using one of two methods: the Oracle Fusion SOA Composer or the Oracle Fusion Middleware Domain Value Map user interface. The SOA Composer is typically considered easier to use.
4. On the Manage Planning Data Collection Processes page, enable the corresponding entity for cross-referencing.
5. Determine an ongoing procedure for adding new values into the domain value map when new values occur for a business object.

Notes regarding usage of the Oracle Fusion SOA Composer:

- Depending upon how the SOA suite is installed at your location, you should have the SOA Designer role assigned in order to edit the data value maps.
- Choose DVM in MscCollConfigurationXReferenceComposite.
- You must save your changes and commit your changes.

- The APS Weblogic server must be bounced before the changes are visible to the collections engine.
- For documentation regarding using the SOA Composer, refer to the Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite document, Using Oracle SOA Composer with Domain Value Maps section.

FAQs for Manage Planning Data Collection Processes

Can I use continuous collection to collect item costs?

The continuous collection data collection method is partially supported for item costs. Item costs are collected in the next incremental collection cycle for previously existing items when one or more item organization attributes in addition to item cost have changed.

When a new item is defined, the item cost for the new item is not collected in the next incremental collection cycle. If an existing item is not changed other than an update to the item cost, the item cost change is not picked up in the next incremental collection cycle.

Tip

If items are added frequently, item costs are changed frequently, or both, then targeted collection of item costs should be routinely performed, perhaps once a day.

Perform Orchestration Data Collections

Loading Data into the Data Collections Staging Tables Using Oracle Data Integrator: Explained

To use the staging tables upload method, you must load the data you extract from your external source systems into the staging tables. You can use Oracle Data Integrator to load the extracted data into the staging tables.

If you have installed Oracle Data Integrator (ODI), and configured ODI for use by Oracle Fusion applications, you can load data to the staging tables by scheduling the Perform Data Load to Staging Tables process, PerformOdiSatagingLoad. To use this process, you must perform these steps and understand these details:

- Steps to use the Perform Data Load to Staging Tables process
- Steps to manually prepare and update the required dat files
- Details regarding the Perform Data Load to Staging Tables process
- Steps to verify execution status after starting the Perform Data Load to Staging Tables process
- Details regarding verifying the Perform Data Load to Staging Tables process execution status
- List of interface ODI scenarios run for each business entity

Steps to Use the Perform Data Load to Staging Tables Process

The Perform Data Load to Staging Tables process invokes an ODI data load. To use this process, follow these steps:

1. Create a data file for each business entity for which you are extracting data from your external source system. The file type for the data files must be dat. Use the sample dat files provided on My Oracle Support as templates. The data in the files you create must conform to the exact formats provided in the sample files.
 - To obtain the sample dat files, see Oracle Fusion Order Promising Data Collections Sample ODI Data Files, document ID 1361518.1, on My Oracle Support <https://support.oracle.com>.
 - You can open the sample dat files in a spreadsheet tool to review the sample data. The sample data shows the minimum required fields for an entity.
2. Place the dat files in the host where the Supply Chain Management (SCM) ODI agent is installed. The dat files must be placed at this specific location: /tmp/ODI_IN.
 - The path for this location is configured for the SCM ODI Agent. The SCM ODI Agent is an ODI software agent that services ODI related client requests. More information about this agent can be found in the ODI product documentation.
 - After ODI is installed, you must use the ODI console to refresh the variables C_LAST_UPDATED_BY and C_CREATED_BY.
3. Schedule the Perform Data Load to Staging Tables, PerformOdiStagingLoad, process.

Steps to Manually Prepare and Update the Required dat Files

You can develop data extract programs to extract data from your external source systems and store the extracted data into the required dat files in the required format. To manually add data to the dat files, follow these steps:

1. Open the applicable dat file in a spreadsheet tool. When you open the file, you will be prompted to specify the delimiter.

Use the tilde character, ~ , for the delimiter.
2. Add any data records you want to upload to the staging tables into the spreadsheet. Data for date type columns must be in the DD-MON-YY date format.
3. Save the worksheet from the spreadsheet tool into a text file.
4. Use a text editor and replace spaces between columns with the tilde character.
5. Verify that every line terminates with a CR and LF (ASCII 000A & 000D respectively.)
6. Upload the dat file to the /tmp/ODI_IN directory where the SCM ODI agent is running. The location is seeded in the ODI topology. Upload (FTP) the dat file in binary mode only.

7. Review the file in vi after the FTP upload to detect junk characters and, if any, remove them.

Details Regarding the Perform Data Load to Staging Tables Process

The Perform Data Load to Staging Tables process invokes the ODI scenario MASTER_PACKAGE that internally invokes all four projects defined in ODI for collections. Each of these four projects invokes various interfaces. Data is loaded from flat files to staging tables for all the business objects enabled for Oracle Fusion 11.1.2.0.0 through Oracle Data Integrator.

The following are specific details for the process:

- Process Name: PerformOdiStagingLoad
- Process Display Name: Perform Data Load to Staging Tables
- Process Description: Collects planning data from flat files and loads to staging tables using Oracle Data Integrator.
- ODI Project Name: SCM_BulkImport
- ODI scenario Name: MASTER_PACKAGE
- SCM Scheduler: SCM_ESS_ODI_SCHEDULER
- Agent URL: your_host_name:your_port_no/oracleodiagent (substitute your host name and your port number)

Steps to Verify Execution Status after Starting the Perform Data Load to Staging Tables Process

To verify the execution status after starting the Perform Data Load to Staging Tables process, perform these steps:

1. The Perform Data Load to Staging Tables process does not log messages to the scheduled processes side. To check for a log message, query the Request_History table using this select statement:

```
Select * from fusion_ora_ess.request_history where requestid=  
<request_id>;
```

- Check the Status column for the overall execution status of the job and the Error_Warning_Detail column for a detailed error message, if any.
2. Check the ODI scenario execution status details in the ODI operator window. The scenario names are listed in the table in the List of Interface ODI Scenarios Run for Each Business Entity section of this document.
3. If log directories are accessible, check the following ODI logs for specific information on ODI scenario execution path:

```
/slot/emsYOUR_SLOT_NUMBER/apmgr/WLS/user_projects/domains/  
wls_appYOUR_SLOT_NUMBER/servers/YOUR_ODI_SERVER_NAME/logs
```

- Diagnostic: for any errors in execution
- Server: for all the logs specific to ODI console
- Agent: for scenario entry and exit and for session ID

Details Regarding Verifying the Perform Data Load to Staging Tables Process Execution Status

When verifying the Perform Data Load to Staging Table process, remember the following:

- No logs will be written at the scheduled processes side. Also, the session id for ODI scenario cannot be found at the scheduled processes side.
- When viewing the process status on the Scheduled Processes page, a Success status does not mean that all the data got into the staging tables successfully. The Success status only indicates that the scenario is launched successfully. Scenario status must be checked from ODI logs.
- You cannot determine the refresh_number generated by ODI for the current process run from the Scheduled Processes page. To obtain the refresh number, you must use this query to query the msc_coll_cycle_status table and check for the ODI collection_channel:

```
Select * from msc_coll_cycle_status order by refresh_number desc;
```

List of Interface ODI Scenarios Run for Each BusinessEntity

One or more interface ODI scenarios are run for each business entity. Each interface scenario maps to one entity. If any interface Scenario fails in ODI, that entity data is not collected to the staging tables. This table lists the business entities and the interface ODI scenarios run within each business entity.

Business Entity	Interface ODI Scenarios
Work-in-Process Requirements	WIP_COMP_DEMANDS_SCEN WIP_OP_RESOURCE_SCEN
Calendars	CALENDAR_SCEN CALENDAR_WORKDAYS_SCEN CALENDARDATES_SCEN CALENDAR_EXCEPTIONS_SCEN CALENDARSHIFTS_SCEN CALENDAR_PERIODSTARTDAYS_SCEN CALENDAR_WEEKSTARTDAY_SCEN CALENDAR_ASSIGNMENTS_SCEN
Demand Classes	DEMAND_CLASS_SCEN
Global Supplier Capacities	GLOBAL_SUP_CAPACITIES_SCEN
Interorganization Shipment Methods	SHIPMENT_METHODS_SCEN
Item Cost	ITEM_COST_SCEN
Item Substitutes	ITEM_SUBSTITUTES_SCEN
Item Suppliers (Approved Supplier List)	ITEM_SUPPLIERS_SCEN
On Hand	ONHAND_SCEN
Organizations	ORGANIZATIONS_SCEN

Purchase Orders and Requisitions	SUPPLY_INTRANSIT_SCEN PO_IN_RECEIVING_SCEN PO_SCEN PR_SCEN
Planned Order Supplies	PLANNEDORDERSUP_SCEN
Resources	RESOURCES_SCEN RESOURCE_CHANGE_SCEN RESOURCE_SHIFTS_SCEN RESOURCE_AVAILABILITY_SCEN
Routings	ROUTING_OPERATION_RESOURCES_SCEN ROUTINGS_SCEN ROUTING_OPERATIONS_SCEN
Sourcing Rules	SOURCING_ASSIGNMENTS_SCEN SOURCING_RULES_SCEN SOURCING_ASSIGNMENTSETS_SCEN SOURCING_RECEIPT_ORGS_SCEN SOURCING_SOURCE_ORGS_SCEN
Subinventories	SUB_INVENTORIES_SCEN
Trading Partners	TRADING_PARTNERS_SCEN TRADING_PARTNER_SITES_SCEN
Units of Measure	UOM_SCEN UOM_CONVERSION_SCEN UOM_CLASS_CONVERSION_SCEN
Work Order Supplies	WORKORDER_SUPPLY_SCEN

Parameters for the Perform Data Load Process: Points to Consider

To perform a data load from the data collection staging tables, you invoke the Perform Data Load from Staging Tables process. When you invoke the process, you provide values for the parameters used by the process

Parameters for the Perform Data Load from Staging Tables Process

When you perform an upload from the staging tables, you specify values for a set of parameters for the Perform Data Load from Staging Tables process including specifying Yes or No for each of the entities you can load. For the parameters that are not just entities to select, the table below explains the name of each parameter, the options for the parameter values, and the effect of each option.

Parameter Name	Parameter Options and Option Effects
Source System	Select from a list of source systems.
Collection Type	<ul style="list-style-type: none"> • Net change Data in the data repository is updated with the data uploaded from the staging tables. <ul style="list-style-type: none"> • Existing records are updated. For example, on hand is updated with current quantity. • New records are added to the data repository. For example, New purchase orders are added to the data repository. • Targeted Existing data in the data repository is deleted and replaced with the data uploaded from the staging tables. For example, a targeted data load for purchase orders will replace all existing purchase order data with the purchase order data from the staging tables.
Group Identifier	Leave blank or select from the list of collection cycle identifiers. Leave blank to load all staging table data for the selected collection entities. Select a specific collection cycle identifier to load data for that collection cycle only.
Regenerate Calendar Dates	<ul style="list-style-type: none"> • Yes You loaded calendar patterns into the staging tables so you need the concurrent process to generate and store individual dates to run. • No You loaded individual dates into the staging tables so you do not need the concurrent process to generate and store individual dates to run.
Regenerate Resource Availability	<ul style="list-style-type: none"> • Yes You loaded resource availability patterns into the staging tables so you need the concurrent process to generate and store individual dates to run. • No You loaded individual dates into the staging tables so you do not need the concurrent process to generate and store individual dates to run.

The parameters presented for the Perform Data Load from Staging Tables process also include a yes-or-no parameter for each of the entities you can collect

using the staging tables upload method. If you select yes for all of the entities, the data collections will be performed in the sequence necessary to avoid errors caused by data references from one entity being loaded to another entity being loaded.

Important

If you do not select yes for all of the entities, you need to plan your load sequences to avoid errors that could occur because one of the entities being loaded is referring to data in another entity not yet loaded. For more information, see the articles regarding order promising or data collections on My Oracle Support at <https://support.oracle.com>.

Collections Cycle Identifier: Explained

The collection cycle identifier is a unique number that identifies a specific data collection cycle, or occurrence. One cycle of a data collection covers the time required to collect the set of entities specified to be collected for a specific data collection method. The collection cycle identifier is then used in statistics regarding data collections, such as the Data Collection Summary report. The collection cycle identifier is also used for a parameter in various processes related to data collections, such as the Purge Staging Tables process and the Perform Data Load process.

This topic explains the population of the collection cycle identifier when you use collecting data from external source systems as follows:

- Web Service Uploads and the Collection Cycle Identifier
- Staging Tables Uploads and the Collection Cycle Identifier

Web Service Uploads and the Collection Cycle Identifier

When you use the Web service upload data collection method, a collection cycle identifier is included as part of the collected data. You can then use the collection cycle identifier to review statistics regarding the Web service collections, or to search for error and warning records written to the data collection staging tables.

Staging Table Uploads and the Collection Cycle Identifier

If you use the Oracle Data Integrator tool to load your extracted data into the data collections staging tables, a collection cycle identifier is created for each load session. Each record loaded into the staging table during the load session will include the collection cycle identifier for that session.

If you populate the data collection staging tables using a method other than the Oracle Data Integrator tool, you must follow these steps to populate the collection cycle identifier.

1. Groupid is to be populated in column refresh_number of each data collections staging table. In one cycle of loading data into the staging tables, the column should be populated with same value. Get the group id value as follows:

```
SELECT ...NEXTVAL FROM DUAL;
```

2. After a cycle loading data into the data collections staging tables, insert a row as follows into table `msc_cycle_status` for that cycle as follows:

```
INSERT INTO MSC_COLL_CYCLE_STATUS
(INSTANCE_CODE, INSTANCE_ID, REFRESH_NUMBER, PROC_PHASE, STATUS,
 COLLECTION_CHANNEL, COLLECTION_MODE, CREATED_BY, CREATION_DATE,
 LAST_UPDATED_BY, LAST_UPDATE_DATE)
SELECT a.instance_code, a.instance_id, :b1, 'DONE', 'NORMAL',
'LOAD_INTERFACE', 'OTHER', 'USER', SYSTIMESTAMP, USER, SYSTIMESTAMP
FROM msc_apps_instances a
WHERE a.instance_code= :b2 ;
:b1 is instance_code for which data is loaded
:b2 is the groupid value populated in column refresh_number in all
interface tables for this cycle
```

Collecting Calendars and Resource Availability: Points to Consider

When you collect calendars and net resource availability from external source systems, you decide whether to collect patterns or individual dates. Order promising requires individual calendar dates and individual resource availability dates to be stored in the order orchestration and planning data repository. If you collect calendar patterns or resource shift patterns, you must invoke processes to populate the order orchestration and planning data repository with the individual dates used by order promising.

You invoke the necessary processes by specifying the applicable parameters when you run data collections. The processes generate the individual dates by using the collected patterns as input. The processes then populate the order orchestration and planning data repository with the individual calendar dates and the individual resource availability dates.

Calendar Collections

When you collect calendars from external source systems, you decide whether to collect calendar patterns or individual calendar dates. Both methods for collecting data from external source systems, Web service upload and staging tables upload, include choosing whether individual calendar dates must be generated as follows:

- The Web service to upload to calendars includes a parameter to run the Generate Calendar Dates process.

You control whether the process will run. If the parameter is set to yes, then after the Web service upload completes, the process will be launched to generate and store individual calendar dates.

- The parameters for the Perform Data Load from Staging Tables process also include a parameter to run the Generate Calendar Dates process.

You control whether the process will run. If the parameter is set to yes, then after the load from staging tables completes, the process will be launched to generate and store individual calendar dates.

- In both scenarios, calendar data is not available while the Generate Calendar Dates process is running.

When you collect calendars from the Oracle Fusion system, the Generate Calendar Dates process is run automatically.

Restriction

Only calendar strings that are exactly equal to seven days are allowed. Calendar strings with lengths other than seven are not collected. Only calendars with Cycle = 7 should be used.

Resource Availability Collections

When you collect net resource availability from external source systems, you decide whether to collect resource shift patterns or individual resource availability dates. Both methods for collecting data from external source systems, Web service upload and staging tables upload, include specifying whether individual resource availability dates must be generated as follows:

- The Web service to upload to net resource availability includes a parameter to run the Generate Resource Availability process.

You control whether the process will run. If the parameter is set to Yes, then after the Web service upload completes, the process will be launched to generate and store individual resource availability dates.

- The parameters for the Perform Data Load from Staging Tables process also include a parameter to run the Generate Resource Availability process.

You control whether the process will run. If the parameter is set to Yes, then after the load from staging tables completes, the process will be launched to generate and store individual resource availability dates.

- In both scenarios, new resource availability data is not available while the Generate Resource Availability process is running.

You cannot collect net resource availability from the Oracle Fusion source system.

Parameters for the Perform Data Collection Process: Points to Consider

To perform a targeted data collection from the Oracle Fusion system, you use the Perform Data Collection process. When you invoke the process, you provide values for the parameters used by the process.

The Perform Data Collection Process

When you perform a targeted collection, you specify the Oracle Fusion source system to be collected from and the organization collection group to collect for. When you invoke the process, the parameters also include each of the fourteen entities you can collect from the Oracle Fusion source system with yes or no for the parameter options. The table below explains the other two parameters.

Parameter Name	Parameter Options
Source System	The source system presented for selection is determined by what system has been defined as the Oracle Fusion source system when the manage source systems task was performed.
Organization Collection Group	The organization collection groups presented for selection are determined by what organization groups were defined when the manage source systems task was performed for the selected source system.

The parameters presented also include a yes-or-no parameter for each of the entities you can collect. If you select yes for all of the entities, the data collections will be performed in the sequence necessary to avoid errors caused by data references from one entity being loaded to another entity being loaded.

Important

If you do not select yes for all of your entities, you need to plan your load sequences to avoid errors that could occur because one of the entities being loaded is referring to data in another entity not yet loaded. For more information, see the articles regarding order promising or data collections on My Oracle Support at <https://support.oracle.com>.

Organization Collection Group: Explained

When you perform a targeted collection from the Oracle Fusion source system, you use an organization collection group to contain the collections processing to only the organizations with data that is needed for the order orchestration and planning data repository. Organization collection groups limit targeted collections from the Oracle Fusion source system to a specific set of organizations.

You perform the following actions for organization collection groups:

- Define an organization collection group.
- Use an organization collection group.

Define an Organization Collection Group

You define organization groups when managing source systems for the source system where the version equals Oracle Fusion. For each organization in the organization list for the Oracle Fusion source system, you can specify an organization group. You can specify the same organization group for many organizations.

Use an Organization Collection Group

You use an organization collection group when you perform a targeted collection from the Oracle Fusion source system and you want to contain the collections processing to a specific set of organizations. You specify which organization

group to collect data from by selecting from the list of organization groups defined for the Oracle Fusion source system. Data will only be collected from the organizations in the organization group you specified.

For example, if only certain distribution centers in your Oracle Fusion source system are to be considered for shipments to your customers by the order promising and order orchestration processes, you could create a DC123 organization group and assign the applicable distribution centers to the DC123 organization group when managing source systems. When you perform a targeted collection for the Oracle Fusion source system, you could select DC123 for the organization collection group.

Review Orchestration Collected Data

Data Collections Daily Monitoring: Explained

When you manage the data collection processes, you use the Process Statistics report and the Data Collection Summary report to routinely monitor your collections. When error records are reported, you query the data staging tables for further details regarding the error records. You can also review most of your collected data using the review collected data pages.

The following information sources are available for you to monitor data collections:

- Process Statistics report
- Data Collection Summary report
- Review collected data pages
- Staging table queries

Process Statistics Report

You view the Process Statistics report to monitor summary of statistic for the daily collections activity for each of your source systems. This report is available on the **Actions** menu when managing data collection processes for either the continuous collection publish process or the collections destination server. The day starts at 00:00 based on the time zone of the collection server.

For the Oracle Fusion source system, statistics are provided for both the continuous collection and the targeted collection data collection methods. For each external source system, statistics are provided for the Web service upload and for the staging tables upload data collection methods. The following statistics are provided in the Process Statistics report:

- Number of collection cycles for the current day
- Average cycle time in seconds
- Average number of records
- Average number of data errors

Note

The process statistics provide summary information, and are not intended for detailed analysis of the collections steps. Use the Oracle Enterprise Scheduler Service log files for detailed analysis.

Data Collection Summaries

You view the Data Collection Summary report to monitor statistics regarding the data collection cycles for each of your source systems. The summary report shows last the results of the last 20 cycles of all collection types. This report is available on the Action menu when managing data collection processes for the collections destination server.

The Data Collection Summary report provides information for each source system. If a source system was not subject to a data collection cycle for the period covered by the summary, an entry in the report states that there are no cycles in the cycle history for that source system. For each source system that was subject to a data collection cycle for the period covered by the summary, the following information is provided for each data collection method and collected entity value combination:

- The data collection method
- The collection cycle number
- The entity collected and, for that entity, the number of records collected, the number of records with data errors, and collection duration
- Time started
- Time ended

Review Collected Data Pages

You can review most of your collected data by using the Review Planning Collected Data page or the Review Order Orchestration Collected Data page. Both pages include a list of entities from which you select to specify the entity for which you want to review collected data. The list of entities is the same on both pages. Most of the entities listed on the review collected data pages are identical to the entities you select from when you run collections, but there are a few differences.

Some of the entities on the list of entities you select from when you review collected data are a combination or a decomposition of the entities you select from when you run collections. For example, the Currencies data collection entity is decomposed into the Currencies entity and the Currency Conversions entity on the review collected data pages. For another example, the Supplies entity on the review collected data pages is a combination of data collection entities including the On Hand entity and the Purchase Orders and Requisitions entity.

A few of the data collection entities cannot be reviewed from the review collected data pages. The data collection entities that are not available for review on the review collected data pages are Resources, Resource Availability, Routings, Work-in-Process Resource Requirements, and Customer Item Relationships.

Staging Table Queries

If errors or warnings have been encountered during data collections, you can submit queries against the staging tables to examine the applicable records. For more information regarding the staging tables and staging table columns, see the articles regarding order promising or data collections on My Oracle Support at <https://support.oracle.com>.

Errors and Warnings When Collecting Data from External Source Systems: How They Are Handled

When you are collecting data from external source systems, the data collection processes perform many data validation checks. If the data validations fail with errors or warnings, the steps taken by the data collection processes vary slightly depending upon whether the Web service upload data collection method or the staging tables upload data collection method is used.

In both cases, records where errors are found are not loaded into the order orchestration and planning data repository. Instead records are loaded into, or remain in, the applicable staging tables with an appropriate error message. Records where only warnings are found are loaded to the data repository, and records are loaded into, or remain in, the applicable staging tables with an appropriate warning message.

Settings That Affect Error Handling When Collecting Data from External Source Systems

The handling of errors and warnings encountered when the data collection processes validate data during collections from external source systems depends upon which data collection method is used, Web service upload or staging tables upload.

How Errors and Warnings Are Handled

When you are running data collections using the Web services method, records without errors or warnings are loaded into the data repository. For records with errors or warnings the following error and warning handling steps occur when using the Web services method:

- **Warnings:** Records with warnings are fixed automatically, then loaded into the data repository and copied into the applicable staging tables with the appropriate warning message.
- **Errors:** Records are loaded to the applicable staging tables instead of the data repository and are marked with the appropriate error message. When there is an error due to missing mandatory fields, in cases where possible, the collections process will retry the record. After several unsuccessful retry attempts, the record will be marked as error. In some cases, retrying is not an option, and the record will be immediately marked as an error.

When you are running data collections using the staging tables upload method, the following error and warning handling steps occur:

- **Warnings:** Records are loaded into the data repository and remain in the staging tables with the appropriate warning message. The message is associated with the record in the data repository.
- **Errors:** When there is an error due to missing mandatory fields, in cases where possible, the collections process will retry the record. After several unsuccessful retry attempts, the record will be marked as error. In some cases, retrying is not an option, and the record will be immediately marked as an error.

Error Handling Example

When a Planned Order Supplies record is collected, many validations occur for which an error is recorded if the validation fails.

For example, the supplier name is validated against the suppliers data in the order orchestration and planning data repository. If the supplier name is not found, the validation fails with an error condition, and the following steps occur:

- The Planned Order Supplies record is not loaded into the data repository.
- The Planned Order Supplies record is loaded into the applicable staging table, or remains in the applicable staging table, with an error message stating invalid supplier or invalid supplier site.

Warning Handling Example

When a Planned Order Supplies record is collected, many validations occur for which a warning is recorded if the validation fails.

For example, the Firm-Planned-Type value in the record is validated to verify that the value is either 1 for firm or 2 for not firm. If the validation fails, the failure is handled as a warning, and the following steps occur:

- The Planned Order Supplies record is loaded into the data repository with the Firm-Planned-Type value defaulted to 2 for not firm.
- The Planned Order Supplies record is also loaded into the applicable staging table, or remains in the applicable staging table, with a warning message stating invalid firm planned type.

Purge Collected Data Processes: Points to Consider

You use the Purge Data Repository Tables process to delete all collected data from the order orchestration and planning data repository that was collected from a specific source system. You use the Purge Staging Tables process to remove data that you no longer need in the data collections staging tables.

The Purge Data Repository Tables Process

You use the Purge Data Repository process to delete all data for a source system from the order orchestration and planning data repository. The process enables you to delete data for a specific source system. You typically use the Purge Data Repository process when one of your source systems becomes obsolete, or when you decide to do a complete data refresh for a set of collection entities.

The Purge Data Repository process has only two parameters, both of which are mandatory. This table explains the two parameters.

Parameter Name	Parameter Options
Source System	Select a source system for the list of source systems. All data for the selected system will be deleted from the data repository.
Purge Global Entities	Yes or No If you select yes, in addition to the applicable data being deleted for the source-specific entities, all data from global entities will also be deleted. If you select no, data will be deleted from the source-specific entities only.

The Purge Staging Tables Process

You use the Purge Staging Tables process to delete data from the data collection staging tables.

The following table explains the parameters you specify when you run the Purge Staging Tables process. In addition to the five parameters explained below, you specify yes or no for each of the twenty-five data collection entities.

Parameter Name	Parameter Options
Source System	Select a source system for the list of source systems. Data will be deleted for this source system only.
Record Type	The record type specifies which type of records to purge as follows: <ul style="list-style-type: none"> • Error Purge only error records. • Warning Purge only warning records. • Retry Purge only records marked as retry. • Complete Purge only records that have been successfully processed and data stored in the data repository. • All Purge all records.
Collection Cycle ID	Specify a value for the collection cycle identifier to purge data for a specific collection cycle only, or leave blank.

From Date Collected	Specify a date to purge data from that date only, or leave blank.
To Date Collected	Specify a date to purge data up to that date only, or leave blank.

FAQs for Review Planning Collected Data

What's an order orchestration reference object?

One of the objects in the set of objects used by the orchestration processes to determine the meaning and descriptions for names or codes, such as payment terms names, freight-on-board codes, and mode-of-transport codes.

The sales order data passed to the orchestration processes contains the names or codes, but the processes need to display the meanings or descriptions. The data to determine the meanings or descriptions for the names or codes must be collected into the order orchestration and planning data repository.

For example, sales order information is passed to the Order Orchestration processes containing a freight-on-board code equal to 65, and the order orchestration and planning data repository contains a record with freight-on-board code equal to 65. The processes use the matching codes to determine that the freight-on-board code meaning is equal to Origin, and the description is equal to Vendors responsibility.

Tip

For the full list of order orchestration reference objects, review collected data for the order orchestration reference objects, and view the list of values for the **Lookup Type** field.

Define Orchestration

Define Orchestration: Overview

Oracle Fusion Distributed Order Orchestration automates order orchestration across fulfillment systems using highly adaptable, flexible business processes. The following setups are required for orchestration:

- **Task type:** Mechanism used to group fulfillment tasks together. Each task type contains a selection of services that communicate with a specific type of fulfillment system, for example, a billing system.
- **Orchestration process definition:** Business process required to fulfill a fulfillment line. It includes the sequence of task layer service calls, as well as planning details, change management parameters, and status conditions.
- **Change logic:** Set of rules that control how changes to booked orders are handled by the orchestration process.

- Process planning: Schedule that shows the completion date of each task and of the orchestration process itself.
- Jeopardy threshold and priority: Mechanism for indicating late-running tasks at runtime.
- Statuses and status conditions: Indicators of the statuses of order and process objects, including task, orchestration process, fulfillment line, orchestration order line, and orchestration order. Status conditions are the rules that determine when any of these order and process objects reaches a particular status.

Manage External Interfaces

Manage External Interfaces: Overview

The external interface layer is the functional component within Oracle Fusion Distributed Order Orchestration that manages the communication between Distributed Order Orchestration and external fulfillment systems. Its primary functions are routing the fulfillment request and transforming the data.

The external interface layer enables loose coupling between Distributed Order Orchestration and fulfillment systems:

- Abstracts external systems from the orchestration process definition to minimize changes when adding new order capture or fulfillment systems.
- Provides an extensible, SOA-enabled framework for flexible integration to external applications.
- Provides a complete, open, and integrated solution that lowers total cost of ownership.

When the setup is done, Distributed Order Orchestration can connect to any fulfillment system.

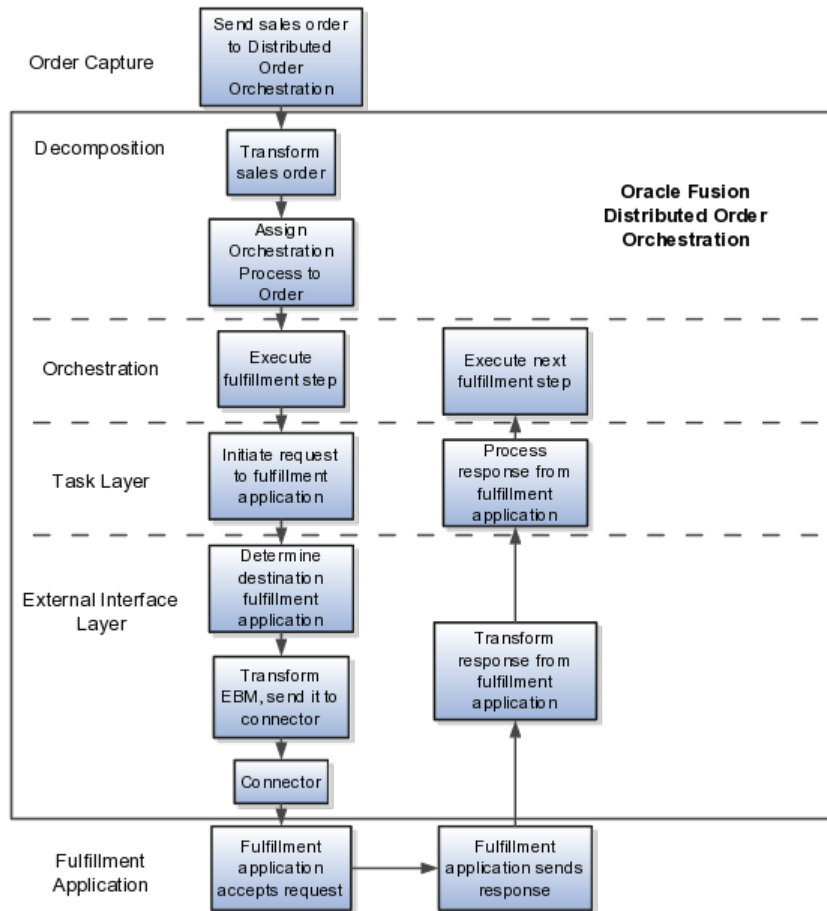
External Interface Layer: How It Works with Integrating External Fulfillment Applications

The external interface layer is a module through which fulfillment requests from Oracle Fusion Distributed Order Orchestration are routed to external fulfillment applications. An external fulfillment application is a fulfillment application that is not an Oracle Fusion application.

This figure shows the flow of a sales order from the order capture application to the fulfillment application. The following steps are illustrated:

1. Send the sales order to Distributed Order Orchestration.
2. Transform the sales order to an orchestration order.
3. Assign an orchestration process to the orchestration order.
4. Execute the fulfillment step.
5. Initiate the request to the fulfillment application.

6. Determine the destination fulfillment application. This is done with routing rules.
7. Transform the enterprise business message (EBM) and send it to the connector.
8. Connector--The EBM is transformed to an application business message (ABM) and is sent to the fulfillment application.
9. Fulfillment application sends a response.
10. Transform the response from the fulfillment application.
11. Process the response from the fulfillment application.
12. Execute the next fulfillment step.



Orchestration

The orchestration layer is the module in which fulfillment tasks are executed. Fulfillment requests originate here and must be communicated to integrated fulfillment applications.

Task Layer

The task layer contains the services to request fulfillment tasks in integrated fulfillment applications. Distributed Order Orchestration provides services for some common fulfillment system functions, such as shipping and invoicing. You

can add task layer functionality through the Template task layer services. The orchestration layer interacts with the external interface layer through the task layer.

External Interface Layer

The external interface layer uses data and routing rules to determine where to route fulfillment requests, and ultimately sends the requests to the appropriate fulfillment systems. Fulfillment requests are processed as follows:

1. The external interface layer receives a request from the task layer.
2. The external interface layer cross-references the data in the request.
3. The external interface layer transforms the request to enterprise business message format.
4. The external interface layer sends the enterprise business message to the connector. The connector is part of the external interface layer. The connector is the responsibility of the implementer.
5. The connector transforms the enterprise business message to an application business message that the fulfillment application can process.
6. The connector calls the fulfillment application.

In this framework, the external fulfillment application does not communicate directly with the orchestration process. The external interface layer acts as an intermediary through which requests are sent. When you integrate a new fulfillment application, you register the connector web service that integrates the fulfillment application with the external interface layer. Because the fulfillment application and the orchestration process are not connected directly, it is possible to implement multiple service providers for fulfillment applications at run time. This framework precludes the need for modifying the orchestration process every time a new fulfillment application is integrated. After the web services and routing rules are configured, Distributed Order Orchestration immediately can interface with the new external fulfillment applications.

External Interface Routing Rules: Explained

Use external interface routing rules to determine to which fulfillment system a fulfillment request must be routed. You can use order, fulfillment line, and process definition attributes to select the fulfillment system connectors. The rules are executed in Oracle Business Rules engine.

Creating External Interface Routing Rules: Examples

Use these scenarios to understand how to use external interface routing rules.

Task Type Determines Routing of Request

You want orchestration orders that are ready to be shipped to go to the shipping fulfillment system. You write an external interface routing rule that requires that if the task type code of an orchestration order is Shipment, then route the request to the ABCShippingSystem connector.

Customer Attribute Determines Routing of Request

Your company has two invoicing systems. When it is time to send out an invoice, you want Widget Company always to be invoiced by system ABC. You write an external interface routing rule that requires that if the customer is Widget Company and the task type code is Invoice, then route the request to ABCInvoicingSystem.

Creating an External Interface Routing Rule: Worked Example

This example demonstrates how to create an external interface routing rule that routes a request to a particular fulfillment system.

Summary of the Tasks

Create the following external interface routing rule: If a request contains server model ABC (Inventory Item ID 100000003357002) and a task type code of invoice, then route the service request to the XYZ fulfillment system.

Navigating to the Manage External Interface Routing Rules Page

1. On your implementation project, expand the Define Sales Order Fulfillment task list.
2. Expand the Define Orchestration task list.
3. Click the **Go to Task** icon next to Manage External Interface Routing Rules.

Creating a New Rule

1. Click the arrow next to the **Add** icon, and select Add Rule. A new rule called Rule 1 appears.
2. Replace Rule 1 with Route Invoice Request Rule.
3. Click the **Expand** icon.
4. Click the Show Advanced Settings arrows.
5. Enter a description, such as: Rule that routes invoice requests for product ABC to XYZ invoice system.
6. Click the **Left Value** icon.
7. Expand DOOExternalInterfaceLayer.
8. Expand FulfillLineTLVO.
9. Select InventoryItemId.
10. Click **OK**.
11. In the Operators list, select the Is list item
12. In the field next to the **Right Value** icon, enter "100000003357002L". L denotes a long value.
13. Click the arrow to display the list, and select Simple Test.

14. In the new row, click the **Left Value** icon.
15. Expand DOOExternalInterfaceLayer
16. Expand HeaderTLVO.
17. Select TaskName.
18. Click **OK**.
19. In the Operators list, select the Is list item.
20. In the field next to the **Right Value** icon enter "Invoice".
21. In the Then region, click the **Add Action** icon.
22. Click the Action list to the right of the check box.
23. Select Assert New from the list. DOOExternalInterfaceLayer.Result appears in the field.
24. Click the **Edit Properties** icon to the right of the down arrow. The Properties window appears.
25. In the **Value** field of resultObj, enter a valid connector name.
26. In the **Value** field of resultObjKey, enter "SERVICE_NAME". The Properties window appears.
27. Click **OK**.
28. Click **Save**.

Creating an External Interface Routing Rule for Fulfillment Systems When Customer Cross-Referencing Data Is Not Found in Oracle Fusion:: Worked Example

This example demonstrates how to create a routing rule that makes it possible for an order to proceed to an integrated fulfillment system when cross-reference data are not found for that system.

Summary of the Tasks

Create the following external interface routing rule: If the task type code on the fulfillment line is Shipment, then route the service request to the Shipping Fulfillment System A.

The customer data of Shipping Fulfillment System A are not found as cross-references in Oracle Fusion, so you must set the value of parameter RESOLVE_XREF_FOR_CUSTOMERINFO to Yes to indicate to the cross-reference service that Oracle Fusion system keys are returned by this service. If cross-referencing data were available, then target fulfillment system keys would be returned.

Note

This rule will send all shipment requests to Shipping Fulfillment System A. If you want to have more than one shipping system, then you must add conditions to differentiate the systems.

Create the If Clause

1. On the Manage External Interface Routing Rules page, click the arrow next to the Add icon and select Add Rule.
2. Enter a rule name, such as the name that was used to register the system connector on the Manage Web Service Details page: "Shipping Fulfillment System A"
3. Near the rule name, click the Expand icon.
4. Next to the rule name, click the Show Advanced Settings icon.
5. Select the Advanced Mode and Tree Mode check boxes.
6. In the Root list, select DOOExternalInterfaceLayer.HeaderTLVO.
7. In the field of the If region, enter: header
8. After "is a" select DOOExternalInterfaceLayer.HeaderTLVO.
9. In the field below Selected Tests, enter "Shipment". You must use quotation marks. You can invert the values in this step and the step in which you entered "header."
10. Select the Equals Ignore Case operator.
11. Click the Right Value icon.
12. Expand DOOExternalInterfaceLayer.
13. Expand HeaderTLVO.
14. Select TaskTypeCode.
15. Click OK.

Create the Then Clause

1. In the Then region, select Assert New.
2. In the list, select DOOExternalInterfaceLayer.Result.
3. Click the Edit Properties icon to the right of the list.
4. Click in the Value field of resultObj.
5. Enter "Shipping Fulfillment System A" (or the connector name you entered on the Manage Web Service Details page). You must use quotation marks.
6. Click in the Value field of ResultObjKey.
7. Enter "SERVICE_NAME". You must use quotation marks.
8. Click OK.
9. Select Assert New. If you are using Fulfillment task layer service, then you do not need to create a new rule but can add this action to the existing routing rule.
10. In the list, select DOOExternalInterfaceLayer.Result.
11. Click the Edit Properties icon to the right of the list.
12. Click in the Value field of resultObj.
13. Enter "Yes". You must use quotation marks.

14. Click in the Value field of ResultObjKey.
15. Enter "RESOLVE_XREF_FOR_CUSTOMERINFO". You must use quotation marks.
16. Click OK.
17. Click Save.

Template Task Layer Mandatory Setup: Explained

Some setup is required to use the template task layer. Some of these setup activities are mandatory because processing cannot occur without the setup information. Other setup activities are optional and will depend on the desired behavior of the services that are associated with the new task type that you are creating. You may set up as many different uses of the template task layer as you need.

The following setup steps are mandatory:

- Create a custom task type.
- Assign status codes to the task type.
- Create the connector.
- Register the connector Web service.
- Use the task type in orchestration process definitions.

Create a Custom Task Type

Create a custom task type on the Manage Task Types page. When you create a custom task type, two services are created, one that corresponds to the Create (outbound) operation code and the other that corresponds to the Inbound operation code. You can specify names for these two services, and you can add services that correspond to the other available operation codes (Cancel, Change, Get Status, Apply Hold, and Release Hold). Create at least one task for each new task type.

Assign a Status Code to the Task Type

Assign status codes to each custom task type. A few system status codes are defaulted, for example, Canceled, Change Pending, Cancel and Pending. The status codes that are associated with each task type also control the values for exit criteria for wait steps that use this task type and for the value of the next expected task status in the orchestration process step definition. You can create new status codes, or you can assign existing status codes to the new custom task type.

Create the Connector

Create the connector that integrates Distributed Order Orchestration with the fulfillment system that will perform the tasks and services of the new task type.

Register the Connector Web Service

Register the Web service that is required for integration to the fulfillment system that will perform the tasks and services of the new task type.

Use the Task Type in Orchestration Process Definitions

You use the new task type and the tasks and services within it by building them into an orchestration process definition, just as you would with the predefined task types, tasks and services. Because splits are allowed for these services, the services may be used only in one branch that is then defined as the single branch that allows services that can be split.

Template Task Layer Optional Setup: Explained

Some setup is required to use the template task layer. Some of these setup activities are mandatory because processing cannot occur without the setup information. Other setup activities are optional and will depend on the desired behavior of the services that are associated with the new task type that you are creating. You may set up as many different uses of the template task layer as you need.

The following setups are optional:

- **Preprocessing service:** You can add preprocessing logic to the actions built into the template task layer service. Your organization might want to add preprocessing logic that defaults data onto the outbound request or validates it.
- **Postprocessing service:** You can add postprocessing logic to the actions that are built into the template task layer service. Your organization might want to add postprocessing logic that defaults logic onto the inbound request, validates the inbound request, or interprets any attributes or messages that are returned by the fulfillment system that might indicate the need for split processing.
- **Change management:** If you want change management processing for orchestration process steps that use one of these task types, then specify the attributes for the task type on the Manage Order Attributes That Identify Change pages. You also need to be sure there are Update and Cancel services and their associated connectors.
- **Define hold codes:** If you want to apply holds to the new services, then create hold codes for them. Hold All applies to the new services, as well as to the existing ones.
- **Define jeopardy thresholds:** If you want to compute specific jeopardy scores on the new custom tasks, then define jeopardy thresholds for them.
- **Define processing constraints:** If you know of some circumstances under which a new custom task should not be called, then define processing constraints to control the behavior.
 - **Define preprocessing validation of mandatory attributes:** You may want to use processing constraints to declare which attributes in the outbound request payload are mandatory.
 - **Define postprocessing validation of mandatory attributes:** You may want to use processing constraints to declare which attributes in the inbound response payload are mandatory.

- Define data set used as part of outbound request: By default, the template task layer uses a complete data structure to communicate Oracle Fusion Distributed Order Orchestration attributes between GetValidFLLineData, the preprocessing service, and the external interface layer routing rules. You can trim the data set to a functionally appropriate size to make processing more efficient.
- Register error messages: If your external fulfillment systems send Distributed Order Orchestration error messages that you want to process and display in the Distributed Order Orchestration messaging framework, then you must register these messages.

Sales Order Attachments: Explained

A sales representative may add attachments while creating a sales order. An attachment might be a document with requirements for manufacturing, a memo for price negotiation, or a URL for product assembling instructions, to name just a few possibilities. Oracle Fusion Distributed Order Orchestration accepts the attachments as part of the sales order. You can view attachments in the Order Orchestration work area and subsequently send them to the necessary fulfillment system. Attachments cannot be sent from the fulfillment system to Distributed Order Orchestration or from Distributed Order Orchestration to an order capture system.

Attachment Configuration: Explained

Sales order attachments can be transmitted from the order capture system to Oracle Fusion Distributed Order Orchestration and from Distributed Order Orchestration to fulfillment systems. To enable transmission of sales order attachments to Distributed Order Orchestration, you must collect the document category during orchestration data collection. To enable transmission from Distributed Order Orchestration, you must invoke the AttachmentsAM public service. Use this service to select and send attachments to the designated fulfillment system, based on the type of the fulfillment request and the category of the attachment.

Web Service Setup: Explained

Web services are used to integrate fulfillment applications with Oracle Fusion Distributed Order Orchestration. Distributed Order Orchestration has a Web service broker that routes requests from the fulfillment task layer to one or more fulfillment systems and vice versa. The following explains how Web services are set up.

1. Create the connector.
2. Deploy the connector.
3. Register the connector.
4. Create external interface routing rules.

Create the Connector

Define an XSLT transformation file to transform the Distributed Order Orchestration fulfillment task message to a Web service-specific message. You can use the Oracle JDeveloper mapper tool or any other tool of choice. Similarly, define the XSLT transformation file to transform the response from the Web service to a message specific to Distributed Order Orchestration.

Deploy the Connector

Make a copy of the connector template, and replace the XSLT transformation files with the files you created for the connector.

Register the Connector

Register the connector on the Manage Web Services page. You must create the source system, so that it is available for selection from this page.

Create External Interface Routing Rules

Create external interface routing rules on the Manage External Interface Routing Rules page. These are the business rules that determine to which fulfillment system requests are routed.

User Credential Key: Explained

The user credential key is a user and password combination created in the credential stores, or Oracle Wallet Manager. This key provides for secure authenticated communication between Oracle Fusion Distributed Order Orchestration and fulfillment systems. You must create a user credential key to integrate Oracle Fusion Distributed Order Orchestration with external services.

Follow the instructions for adding a key to a credential map in the Oracle Fusion Middleware Security Guide 11g Release 1 (11.1.1). You must have the administration privilege and administrator role. In the Select Map list, select oracle.wsm.security. Enter the key, user name, and password from the service that is being integrated with Oracle Fusion Distributed Order Orchestration. Register the user credential key on the Manage Web Service Details page of Distributed Order Orchestration.

Web Service Invocation from Oracle Fusion Distributed Order Orchestration: Explained

To communicate with an external fulfillment system, Oracle Fusion Distributed Order Orchestration invokes user-defined web services through the external interface layer. The external interface layer is a module through which fulfillment requests from Oracle Fusion Distributed Order Orchestration are routed to external fulfillment applications. The external interface layer is not

integrated directly with the orchestration layer, so fulfillment applications can be integrated dynamically without disrupting orchestration processes. To integrate an external fulfillment application, you can create a new connector web service or use an existing service. You log the web service URL on the Manage Web Service Details page of Distributed Order Orchestration. Then you create business rules to select the web service. Because the external interface layer is an intermediary layer that uses open access (mainly HTTP), extra security setup is required. You must ensure that the user credential is valid in the target system and in the security certificate to encrypt and decrypt messages.

User Credential

Obtain a user credential key in Oracle Wallet Manager, and add it to the invoking server's identity store.

A user credential is a user name and password defined in the target system and is used for authenticating incoming requests. This means that the consumer of the service must pass in these credentials as part of the request.

Ask the service provider for the user credentials to be used when invoking their service. The IT administrator must add the user credentials provided by the service provider to the service consumer's server and provide a reference, which is called a CSF-KEY.

Register the external system in the Manage Source System Entities flow. For each service hosted on the external system that you plan to interact with, register the service on the Manage Web Service Details page. Provide a name (Connector Name) for the service, the physical location (URL) of the service; and the CSF-KEY pointing to the user credential that will be used when interacting with the external service. Because there is no requirement to match the actual name of the connector you can provide a short name, which can then be used in routing rules. The CSF-KEY applies to all services offered by the target system.

Security Certificate

Oracle recommends that you configure servers that are running external Web services that must be invoked to advertise the security certificate in the WSDL. The default setting in Oracle WebLogic Server is to advertise the security certificates. Check whether your servers support this feature; if so, then enable the feature.

If you cannot set up the server this way, then use the keystore recipient alias. Ask the service provider for the security certificate. An IT administrator imports the target server security certificate into the invoking server and provides a reference, which is called a keystore recipient alias. Add this alias to the external service entry that was created when you specified the user credential. Register this keystore recipient alias on the Manage Web Service Details page against the records created for that system. This key applies to all services offered by the target system.

If the other options do not work, then configure the target servers to use the Oracle security certificate. Import this certificate into your target servers. No setup is required on the invoking server for the security certificate.

Define Orchestration Processes

Orchestration Process Definitions: Explained

An orchestration process is a process flow that approximates your organization's fulfillment process. An orchestration process contains a sequence of steps that takes a fulfillment line through the fulfillment process. An orchestration process contains the instructions for how to process an order, such as which steps and services to use, step dependencies, conditional branching, lead-time information, how to handle change orders, which status values to use, and more. You define orchestration processes according to your organization's needs. You create rules, so that at run time the appropriate orchestration process is automatically created and assigned to the new fulfillment lines.

If you want to use the ShipOrderGenericProcess and ReturnOrderGenericProcess predefined orchestration processes, then you must generate them and deploy them; it is not necessary to release them.

Defining Orchestration Processes: Examples

Orchestration process definitions include the sequence of task layer service calls, as well as planning details, change management parameters, and status conditions. Use the following examples to understand how you can use orchestration process definitions to model your business processes.

Sequence of Task Layer Service Calls

You are an order administrator at a company that sells widgets. You list the logical steps that are required to fulfill an order. Then you create an orchestration process that mirrors your business process.

The first steps are:

- Step 1: Plant Acknowledgement
- Step 2: Assemble
- Step 3: Wait for Assemble COMPLETE
- Step 4: Ship
- Step 5: Wait for SHIPPED

(Statuses are represented in all uppercase letters.)

Your company requires that a representative call the customer when an invoice exceeds \$100,000. You continue creating the orchestration process by adding a conditional step (which is not a task layer service call):

- Step 6: Conditional node where the orchestration process branches

Afterwards, the steps continue as follows:

- Step 7: Call Customer
- Step 8: Wait for Call Customer COMPLETE

- Step 9: Send High Value Invoice
- Step 10: Wait for High Value Invoice BILLED

Otherwise, the steps are:

- Step 7: Invoice
- Step 8: Wait for BILLED

A merge node ends the branch.

Planning Details

You are an order administrator at a company that sells carpet. Your company has established lead times that enable representatives to monitor the fulfillment process and determine when orders will be fulfilled.

- Schedule: 2 days
- Reservation: 1 day
- Shipment: 6 days
- Invoice: 2 days

You create an orchestration process that contains this information by adding the default lead time to each orchestration process step. When a step is delayed, a background process automatically triggers replanning and expected completion dates are reset.

Status Definitions

You are an order administrator at a company that sells carpet. You have an important customer who requires that you notify the receiving clerk one day before the carpet is shipped. You create an orchestration process for this customer's orders. You use the Carpet Process orchestration process class, which contains the statuses: SHIPPED, RESERVED, READY TO SHIP, SHIPPED, INVOICED. On the Orchestration Process Status tab, you create status conditions for the orchestration process for the special customer, such as: If the status of the CreateShipment step is PRESHIP READY, then use the READY TO SHIP status to represent the status of the orchestration process. Now, the order manager can see in the Order Orchestration work area when the orchestration process status is READY TO SHIP.

Task Types: Explained

A task type is a grouping of services that perform fulfillment tasks. Task types represent common business functions that are required to process an order from its receipt from the order capture application to its execution in the fulfillment application. The following task types are provided by default: Schedule, Reservation, Shipment, Activity, Invoice, Return, and Pause. You can create additional task types by using the Custom and Activity task types. Task types are made up of services and tasks. Service refers to an internal Web service that communicates with the task layer. A task is a representation of the services of a task type. Tasks and services are the building blocks of an orchestration process.

Seeded task types are read-only. You cannot delete task types. You can change the names of task types you create, but it is not recommended.

You can edit the service names of the Activity and Custom task types. You can add services from the pool of available services, but you cannot edit or delete services for custom task types.

Use tasks to represent the services that belong to a task type. For example, you can define a Ship Goods task to represent services from the Shipment task type. When one of the Shipment services is running, Ship Goods appears in the Order Orchestration work area, regardless of whether the Create Shipment or Update Shipment service is called; the services do not appear in the Order Orchestration work area. You can define several tasks for a task type to represent your real-world user tasks, such as ShipGoods or ShipWidgets. Both tasks and services appear in the orchestration process definition.

Task Type Management

Task type management is the registration of internal service references for the task layer.

Creating an Orchestration Process Definition: Worked Example

This example demonstrates how to create an orchestration process definition.

Summary of Steps

The steps for creating an orchestration process definition with branching can be summarized as follows:

1. Enter the required information in the header.
2. Create the orchestration process definition steps.

Enter the Required Information in the Header

On the Manage Orchestration Process Definitions page, click the Create icon.

1. On the Create Orchestration Process Definition page, enter an orchestration process name without any spaces, such as OrchestrationProcess1.
2. Select the Ship Order process class.
3. Select the Common set ID.

Create the Orchestration Process Definition Steps

Create the Schedule Product Step

1. On the Step Definition tab, click the Add Row icon.
2. Next to Step 100, enter a step name, such as Schedule Product.
3. Select the Service step type.
4. Select the Schedule task type.

5. Select the Schedule task name.
6. Select the Create Scheduling service name.

Create the Reserve Product Step

1. With your cursor on the Schedule Product step (step 100), click the Add Row icon.
2. Next to Step 200, enter a step name, such as Reserve Product.
3. Select the Service step type.
4. Select the Reservation task type.
5. Select the Reserve task name.
6. Select the Create Inventory Reservation service name.

Create the Ship Product Step

1. With your cursor on the Reserve Product step, click the Add Row icon.
2. Next to Step 300, enter a step name, such as Ship Product.
3. Select the Service step type.
4. Select the Shipment task type.
5. Select the Ship task name.
6. Select the Create Shipping service name.

Create the Wait for Product Shipment Step

1. With your cursor on the Ship Product step, click the Add Row icon.
2. Next to Step 400, enter a step name, such as Wait for Product Shipment.
3. Select the Service step type.
4. Select the Shipment task type.
5. Select the Ship task name.
6. Select the Wait for Shipment service name.
7. Set the exit criteria to Shipped.

Create the Create Invoice Step

1. With your cursor on the Wait for Product Shipment step, click the Add Row icon.
2. Next to Step 500, enter a step name, such as Create Invoice.
3. Select the Service step type.
4. Select the Invoice task type.
5. Select the Invoice task name.
6. Select the Create Billing Lines service name.

Create the Wait for Invoice Step

1. With your cursor on the Create Invoice step, click the Add Row icon.

2. Next to Step 600, enter a step name, such as Wait for Invoice.
3. Select the Service step type.
4. Select the Invoice task type.
5. Select the Invoice task name.
6. Select the Wait for Billing service name.
7. Set the exit criteria to Billed.

Creating an Orchestration Process Definition with a Pause Task: Worked Example

This example demonstrates how to create a Pause task as part of an orchestration process definition. A Pause task temporarily stops an orchestration process to wait until a particular date or for a particular event. You might want to use a Pause task to wait until all payments are made before releasing an item from layaway or to wait for a specific date to arrive before releasing pre-orders.

Summary of the Tasks

Create an orchestration process for pausing pre-ordered items before shipping.

1. Create an orchestration process definition.
2. Create the Pause task.
3. Create the Start-After condition. The Start-After condition is a rule that outlines when to pause the orchestration process. After the condition is met, the orchestration process proceeds to the next task. You can reevaluate the rule after the condition is met by selecting the reevaluate option. The rule used in this example is: If the customer demand class is Bronze, then release the pause using a scheduled process that searches for the eventName MY_EVENT (not related to Oracle Fusion Middleware event framework). Note that you will create two rules. One rule pauses the task if the conditions are true. The other rule skips the pause if the conditions of the first rule are false.

Creating an Orchestration Process Definition

1. On the Manage Orchestration Process Definitions page, click the Create icon.
2. In the Process Name field in the header, enter a process name. The name cannot have spaces.
3. In the Set list, select a set, such as Common.
4. In the Process Class list, select a class, such as Ship Order Class.
5. On the Step Definition tab, click Add Row.
6. Create the Schedule Goods step.
7. Select the Service step type.
8. Select the Schedule task type.
9. Select the Schedule task name.

10. Select the Create Scheduling service name
11. Click Save.
12. Using the same procedure, create the following additional steps: Create Reservation, Create Shipment, Wait for Shipment Advice, Create Invoice, Wait for Invoice. Select the Service step type for all. Pick a task type that matches the step, for example, select the Invoice task type for the invoicing steps.

Creating the Pause Step

1. Place your cursor in the Create Reservation step, and click the Add Row icon. The new step is added below the step where your cursor is positioned.
2. In the Step Name field, enter Pause.
3. In the Step Type list, select Pause.
4. In the Task Type list, select Pause.
5. In the Task Name list, select Pause.
6. In the Service Name list, select Pause Process.
7. Click Save.

Creating the Start-After Condition

First Rule: Create the If Clause

1. In the Start-After Condition column, select Click for Rule.
2. Next to the rule name, click the Expand icon.
3. Click the Collapse Pane icon to expand the visible space.
4. Click the Left Value icon.
5. In the Condition Browser, expand DooSeededOrchestrationRules.
6. Expand DOOFLine.
7. Select demandClassCode.
8. Click OK.
9. Select the Is operator.
10. Click the Right Value icon.
11. Enter "Bronze". Quotation marks are required.

First Rule: Create the Then Clause

1. In the Then region, select Assert New.
2. Select DooSeededOrchestrationRules.SacResult.
3. Click the Edit Properties icon. The Properties window appears.
4. In the reevaluateFlag field, enter "N". Quotation marks are required.
5. Click the Value icon next to sacType.
6. Expand DooSeededOrchestrationRules.

7. Expand SacResult.
8. Select SAC_TYPE_EVENT.
9. Click OK.
10. Next to the eventName field, enter "MY_EVENT" into the Value field. Quotation marks are required.
11. Click OK.
12. Click Save.

Second Rule: Create the If Clause

1. In the Start-After Condition column, click Click for Rule.
2. Next to the rule name, click the Expand icon
3. Click the Collapse Pane icon to expand the visible space.
4. Click the Left Value icon.
5. In the Condition Browser, expand DooSeededOrchestrationRules.
6. Expand DOOFLine.
7. Select demandClassCode.
8. Click OK.
9. Select the Isn't operator.
10. Click the Right Value icon.
11. Enter "Bronze". Quotation marks are required.

Second Rule: Create the Then Clause

1. In the Then region, select Assert New.
2. Select DooSeededOrchestrationRules.SacResult.
3. Click the Edit Properties icon. The Properties window appears.
4. In the reevaluateFlag field, enter "N".
5. Click the Value icon next to sacType.
6. Expand DooSeededOrchestrationRules.
7. Expand SacResult.
8. Select SAC_TYPE_IMMEDIATE.
9. Click OK.
10. Click OK.
11. Click Save.

Change Processing: How It Processes Changes that Occur During Order Fulfillment

During fulfillment of an order, changes can originate from a variety of sources, such as from the customer through the order capture application or by the order

manager in the Order Orchestration work area. Oracle Fusion Distributed Order Orchestration processes changes automatically, but you can influence some aspects of change processing through some of the setup options.

Settings That Affect Change Processing

Change processing occurs according to certain settings. The following parameters are set at the orchestration process level:

- **Order attributes that identify change:** Attributes that, when changed, require compensation of the orchestration process. Changes to certain attributes do not always require compensation of the orchestration process. For example, an addition of a suite number to a customer's address might not require adjustments to the orchestration process. You can refine how a change is processed for one of these attributes through compensation patterns for orchestration process steps.
- **Processing constraints:** Rules that control attempted changes to an order, for example, what can be changed, when, and by whom. Changes that are forbidden by processing constraints are disallowed, and an error message appears.
- **Change mode:** Setting that determines how frequently state snapshots are taken during an orchestration process. Snapshots are compared during compensation of the orchestration process. For example, if you select Simple, then only the beginning and end states are compared. If you select None, then all changes to the orchestration process are rejected.
- **Cost of change rule:** Rule that specifies the costs to the business due to a requested change.

The following parameters are set at the orchestration process step level:

- **Use transactional item attributes:** Indicates whether the product-specific transactional item attributes need to be considered for change.
- **Use flexfield attributes:** Indicates whether flexfield attributes need to be considered for change.
- **Compensation pattern:** Rule that specifies the adjustments to make when an order is changed. For example, if Distributed Order Orchestration receives a change order with a new warehouse for the Create Shipment step, then Distributed Order Orchestration runs the Cancel service and Create service again. If a compensation pattern is not designated for a process step, then the default compensation pattern is used, which is to run the Update service (or the Cancel and Create services).

How Changes Are Processed

When a change order is received from an order capture system, Distributed Order Orchestration performs a lookup to determine whether the order key has been received before. Distributed Order Orchestration sends a request to the fulfillment system that is responsible for the task that was running when the change order was received. The request has several components: Hold current processing, designate whether a change can be accepted, and send the current status.

Change orders are decomposed and orchestrated in the same manner as new orders. If rules were set up for special processing of change orders, then the rules are applied at this time.

Distributed Order Orchestration checks for header-level processing constraints that prevent change processing. If change processing is allowed, then the delta service is called. The delta service checks the attributes that indicate whether the change must be compensated. If the change requires compensation, then compensation begins after line-level attributes are checked.

Distributed Order Orchestration checks line-level processing constraints. If constraints are violated for even one fulfillment line, then the entire change order is rejected.

When an action on the Order Orchestration work area requires change processing, all the above actions occur except decomposition. After the changes are identified by the delta service, Distributed Order Orchestration analyzes and compensates the process, step-by-step, analyzing the state of each step to determine what processing changes are needed to incorporate the changes to the order. To determine the steps to compensate, Distributed Order Orchestration uses the process step state snapshots taken at each task layer service invocation while the orchestration process was running.

The process delta service identifies all orchestration process steps that are associated with delta attributes. You can opt for the default behavior (context-based undo or update) or specify a business rule that determines the appropriate action as the compensation pattern for each process step. Distributed Order Orchestration evaluates the compensation pattern identified for the step to identify what processing to run in a change scenario. Compensation patterns include undo, redo, update, cancel, and none. The default compensation sequence is first in, first out, based on the orchestration process sequence. If the entire order is canceled, then a last in, first out sequence is used. After the compensating services are completed, processing continues using the original orchestration process specification or the appropriate orchestration process dictated by the changes is started. Expected dates are replanned, unless the entire order is canceled. At this point, change processing is completed.

Order Attributes That Identify Change: Points to Consider

Order attributes that identify change are attributes that, when changed by the order capture application or Order Orchestration work area user, require compensation of an orchestration process step. A change to any one of these attributes requires compensation of a step if the attribute is assigned to the task type associated with the step. For example, if the quantity of a sales order is increased, then additional supply must be scheduled and shipped. The Schedule and Shipment steps of the orchestration process are affected because the quantity is an attribute assigned to those task types.

Attribute Selection

Select an attribute from the list of entities: Fulfillment line, orchestration order line, or orchestration order. Selection of this attribute means that at run time, when a change order is received, the application searches for this attribute on

the entity that you associated it with to determine whether it is was changed on the change order. For example, if you select Scheduled Ship Date on the orchestration order line, then when a change order is received, the application compares the Scheduled Ship Date attribute on the line of the change order with the Scheduled Ship Date attribute of the most recent version of the orchestration order line.

By default, the application searches for a set of attributes, which are indicated by selected check boxes. You cannot deselect them; you can only add more attributes.

If you want flexfield attributes or dynamic attributes associated with specific products to be considered for change, then select Use Flexfield Attributes and Use Dynamic Attributes on the orchestration process definition. You cannot select these attributes individually.

Task Type Selection

The task type selection defines the attributes that will be used to evaluate whether a step using that task type requires compensation. Attributes are predefined for predefined task types, but additional attributes can be added. When you add a new task type, no attributes are defaulted. The task is not evaluated to determine compensation requirements unless you set up these attributes first.

Click **Add All** to add attributes to all existing task types.

Status Values: Explained

Status denotes the progress of an order from beginning to completion. The status of an orchestration order is determined by the status of its fulfillment lines, orchestration processes, and tasks. Status values appear in the Order Orchestration work area, where order managers can use them to quickly understand the progress of an orchestration order or its components.

You create a list of all the statuses that can be used in Oracle Fusion Distributed Order Orchestration. For each status code, you create a display name, which is how the status will appear in the Order Orchestration work area. Then, using the list of defined statuses, create a separate list of statuses that an administrator is allowed to use for each of the following: Fulfillment lines, task types, and orchestration processes. When administrators create status conditions for orchestration processes, they can choose from these status values only. You must define the status values in the Manage Status Values page to make them available for selection when creating status conditions.

Fulfillment Line Status: How It Is Determined

During processing of a fulfillment line, the tasks of the assigned orchestration process are fulfilled step by step. You can determine the status that will be assigned to a fulfillment line at each stage of the process. For example, you can indicate that if a Schedule Carpet Installation task has a status of Pending Scheduling, then the fulfillment line status will be Unscheduled.

Settings That Affect Fulfillment Line Status

You can designate the statuses that represent a fulfillment line when you define an orchestration process. These statuses are used to represent the status of the fulfillment line throughout the application. You create a status rule set that lists a sequence of the statuses that will be attained during the orchestration process and the conditions under which they are assigned to the fulfillment line. For example, you could designate the status Scheduled to be used for the fulfillment line when the Schedule Carpet task reaches a status of Completed.

How Fulfillment Line Status Is Determined

At run time, the application evaluates each of the status conditions sequentially. The true condition with the highest sequence number determines the status of the fulfillment line.

Split Priority: Explained

When an orchestration process splits, two or more instances of the same task result. Split priority is a ranking that is used to evaluate multiple instances of a task that splits. The ranking determines which task status represents the status of the orchestration process. A lower number corresponds to a higher rank. The status with the lower number is used to represent the status of the orchestration process.

For example, an orchestration process splits and results in two instances of the Schedule task. One Schedule task has a status of complete, and the other has a status of pending. Because pending has a split priority of two and complete has a split priority of three, pending is used to represent the status of the orchestration process.

Orchestration Order Status: How It Is Determined

An orchestration order can have one or more orchestration order lines, each in its own status. The status of the orchestration order is based on the orchestration order lines that are mapped to it

How Orchestration Order Status Is Determined

The following table shows how the orchestration order status is determined, given the statuses of the associated orchestration order lines.

Orchestration Order Line Statuses	Orchestration Order Status
All orchestration order lines are completed.	Closed
All orchestration order lines are not completed.	Open
Some, but not all, orchestration order lines are completed.	Partial
All orchestration order lines are canceled.	Canceled

Some orchestration order lines are canceled.	Ignore canceled orchestration order lines, and determine status based on the open orchestration order lines.
--	--

For example, if no orchestration order lines are completed, then the orchestration order status is open.

Orchestration Order Line Status: How It Is Determined

An orchestration order line can have one or more fulfillment lines, each with its own status. The status of the orchestration order line is based on the fulfillment lines that are mapped to it.

How Orchestration Order Line Status Is Determined

The following table shows how the orchestration order line status is determined, given the statuses of the associated fulfillment lines.

Fulfillment Line Statuses	Orchestration Order Line Status
All fulfillment lines are completed.	Closed
All fulfillment lines are not completed.	Open
Some, but not all, fulfillment lines are completed.	Partial
All fulfillment lines are canceled.	Canceled
Some fulfillment lines are canceled.	Ignore canceled fulfillment lines, and determine status based on the open fulfillment lines.

Jeopardy Priorities: Explained

Jeopardy priority indicates the level of risk associated with the delay of a task of an orchestration process. It appears in the Order Orchestration work area as Low, Medium, and High.

Create a jeopardy priority by mapping a jeopardy score range to one of the three severity levels. For example, you could map the jeopardy priority Low to a minimum jeopardy score of 0 and a maximum jeopardy score of 100. Jeopardy priorities are provided by default. You can change the values in the ranges to meet your business needs. You cannot delete or add priorities, or change jeopardy priority names; only Low, Medium, and High are available.

Jeopardy Score: Explained

Jeopardy score is a numerical ranking associated with a delay in the completion of a task of an orchestration process. Jeopardy score indicates how severe a delay is deemed. The jeopardy score is mapped to jeopardy priorities of Low, Medium, and High, which appear in the Order Orchestration work area. The indicator provides a quick visual cue to order managers, so that they can take appropriate action to mitigate a delay.

You determine jeopardy score when you create jeopardy thresholds.

Assignment to Tasks

Jeopardy score is assigned to tasks based on jeopardy thresholds. When a task is delayed, the difference between the required completion date and the planned completion date is calculated. Then the application searches for a threshold that applies to the most number of entities of the task. It searches for a threshold in the following order:

1. Combination of all four elements: Process name, process version, task name, and task type.
2. Process name, process version, and task name.
3. Process name and task name.
4. Process name, process version, and task type.
5. Process name and task type.
6. Task name.
7. Process name and process version.
8. Process name.
9. For task type.

The application searches for a threshold that applies to all four entities of the task: Task type, task name, process name, and process version. If a threshold for that combination is not found, then the application searches for a threshold that applies to the process name, process version, and task name of the task, and so on. After an appropriate threshold is located, the score dictated by the threshold is assigned to the task.

Appearance in Order Orchestration Work Area

The jeopardy priority that appears in the Order Orchestration work area maps back to the task with the highest jeopardy score. In other words, if multiple tasks are in jeopardy within an orchestration process, then the highest jeopardy score is used to represent the jeopardy of the orchestration process. For example, in an orchestration process called Carpet Processing, insufficient supply in the warehouse causes several tasks to be delayed, including the Deliver Carpet task and the Invoice Carpet task. A three-day delay to the Deliver Carpet task maps to a jeopardy score of 100 and a jeopardy priority of Medium; a three-day delay to the Invoice Carpet task carries a jeopardy score of 200 and a jeopardy priority of High. Two hundred is the higher score, so this task's jeopardy score is used to represent the jeopardy of the Carpet Processing orchestration process. In the Order Orchestration work area, this orchestration process displays a jeopardy priority of High.

Task Status Mappings: Explained

Fulfillment tasks have predefined status codes. You can choose to display different status codes from the predefined ones by mapping the predefined status codes to your preferred status codes. The status codes that you change them to appear in the Order Orchestration work area and in other status management areas, as well, such as the Status Conditions tab of an orchestration process definition.

Jeopardy Thresholds: Explained

Jeopardy thresholds are used to monitor and measure orchestration processes. Jeopardy thresholds are ranges of time that a task is delayed. You define a set of ranges for each task of an orchestration process and then assign a score that indicates the severity of the delay. These setups are used to create indicators that appear on the Order Orchestration work area. These indicators help order managers to quickly see the severity of a delay, enabling them to take appropriate action.

When an orchestration process is assigned to an orchestration order, the process is planned taking into account the lead time of steps in the orchestration process and certain key dates from the sales order, such as required completion date. Each task of the process has a planned start and completion date. When a task of the orchestration process is delayed, the whole process is replanned. When a task in the process is expected to be completed after the required completion date of the task, a jeopardy score is automatically assigned to each task based on the jeopardy thresholds you define.

You can define jeopardy thresholds for any combination of the following:

- Task type
- Task name
- Process name
- Process version

You are not required to choose any of the above options. If you leave them at their default setting of All, then the jeopardy threshold applies to all tasks.

Prerequisites

If you want to apply the threshold to a task or orchestration process, then orchestration processes, tasks, and task types must be defined so that you can select them when creating jeopardy thresholds.

Orchestration Process Definition: Points to Consider

Orchestration process definitions contain the information that is used to create an orchestration process at run time. When defining an orchestration process, your choices affect how a fulfillment line is processed to completion.

Oracle Fusion Distributed Order Orchestration provides the following predefined orchestration processes:

- Ship order
- Return order

The Ship Order orchestration process contains the following sequential tasks:

1. Schedule
2. Reservation

3. Shipment
4. Invoice

The Return Order orchestration process contains the following sequential tasks:

1. Return Receipt
2. Invoice

Prerequisites

Before you define an orchestration process, perform the following prerequisite tasks:

- Execute mandatory tasks in Functional Setup Manager.
- Define any additional task types and their associated tasks and services.
- Define any additional status codes and how they will be used for task types, fulfillment lines, and orchestration processes.
- Define any subprocesses that will be used in the process to be defined.
- Define status catalogs that will be used for status conditions. Define the catalogs in Oracle Fusion Product Model, Oracle Fusion Product and Catalog Management, or Oracle Fusion Product Hub.

Header

The header contains basic information that applies to the entire orchestration process. During step definition, you will determine the information that applies to individual steps.

- **Orchestration process class:** This value is required. It is a set of statuses that can be used for this orchestration process.
- **Change mode:** When a change order enters the system, the delta service analyzes the state of each step to determine what processing changes are needed. After this analysis, compensation occurs on the necessary steps. Your selection determines how often snapshots of the orchestration process are taken.
 - **None:** Snapshot of the orchestration process is not taken, and change is not allowed.
 - **Simple:** Snapshot is taken when orchestration process starts and at step where change order is received.
 - **Advanced:** Snapshots are taken at each orchestration process step.

Caution

If you used the Functional Setup Manager migration tool to port test instance data to a production environment, then do not change the process name in either instance. Changing the name could prevent references to other data in the orchestration process from being updated.

Orchestration Process Classes: Explained

An orchestration process class is a set of statuses that can be used for an orchestration process. Use orchestration process classes to simplify orchestration process definition. You can assign the complete set of statuses to any number of orchestration process definitions without having to list the statuses one by one. You do not have to use all the status values in the orchestration process class.

When an orchestration process class is assigned to an orchestration process, you can use only the statuses in that class. The status values that are defined in the orchestration process class are only for the statuses at the orchestration process level, not for the tasks or fulfillment lines.

Status Catalog: Explained

Your organization might need for different fulfillment lines within the same orchestration process to have different status progressions. For example, a model with shippable fulfillment lines and nonshippable fulfillment lines may require different statuses for each type of fulfillment line. A status catalog provides a means to group like items, so that they can achieve the same statuses at the same time. Status catalogs are defined in Oracle Fusion Product Model.

Cost of Change: Explained

Cost of change is a numerical value that represents the impact of a change to an orchestration process. Cost could be the monetary cost to a company or the difficulty associated with the change. This value is calculated and returned to the order capture application, so that the customer service representative can understand how costly it is to make the customer's requested change. The cost of change value can be requested by the order capture application before a change order is submitted to determine whether it should be submitted. Cost of change is calculated also after compensation of a change order is completed. Cost of change is most often used in a postfactor analysis to change practices or processes, or in a business-to-business negotiation.

You assign the cost of change to the orchestration process using a business rule. When the order capture application requests a cost of change determination, the value is calculated and returned, but it is not stored. If you choose not to use cost of change values, then zero is the value used to calculate the cost of change when a change order is submitted.

Creating Cost of Change Rules: Worked Example

This example demonstrates how to create a cost of change rule for an orchestration process, so that order managers are aware of how costly to the company certain changes are. The order administrator of a flooring company wants a few rules that indicate the cost to the company if a change is requested when the fulfillment line is at a certain status. The cost of change is low if the fulfillment line is in Scheduled status, and it is much higher if the fulfillment line is in Shipped status.

Note

The following is an example of a simple rule, which is well suited for rules for an orchestration process with a single line. If you want to write a rule for an orchestration process that has multiple lines, then use advanced mode rules. For more information, see Oracle Fusion Middleware User's Guide for Oracle Business Rules.

Summary of the Tasks

Create If and Then statements for the following rules:

- If the fulfillment line status value is Shipped, then the cost of change is 50.
- If the fulfillment line status value is Scheduled, then the cost of change is 5.

Creating the If Statement of the First Rule

Create the If statement: If the fulfillment line status value is Shipped.

1. On the header of the Orchestration Process Definition page, click **Click for Rule** next to Cost of Change.
2. Click the **New Rule** icon.
3. Optionally, enter a rule name.
4. Click the arrows to expand the rule.
5. Click the **Left Value** icon to search for the left value.
6. Expand DOOFLine.
7. Select Status Value.
8. Click **OK**.
9. Select Is.
10. In the field to the right of your Is selection, enter "SHIPPED". You must surround the text with quotation marks.

Creating the Then Statement of the First Rule

Create the Then statement: Then the cost of change is 50.

1. Click **Insert Action**.
2. Select Assert New.
3. Select Result.
4. Select the **Edit Properties** icon.
5. In the **Value** column of resultObjKey, enter 50.
6. Click **OK**.

Creating the If Statement of the Second Rule

Create the If statement: If the DOO Fulfillment Line status value is Scheduled.

1. Click the **New Rule** icon.
2. Click the arrows to expand the rule.
3. Click the **Left Value** icon to search for the left value.
4. Expand DOOFLine.
5. Select Status Value.
6. Click **OK**.
7. Select Is.
8. In the field to the right of your "is" selection, enter "SCHEDULED". You must surround the text with quotation marks.

Creating the Then Statement of the Second Rule

Create the Then statement: Then the cost of change is 5.

1. Click **Insert Action**.
2. Select Assert New.
3. Select Result.
4. Select the **Edit Properties** icon.
5. In the **Value** column of resultObjKey, enter 5.
6. Click **OK**.
7. Click **Save**.

Creating Line Selection Rules: Worked Example

This example demonstrates how to create a line selection rule that determines which lines to process for a particular step in a case where not all lines should be processed by that step. The order administrator of a company that sells DVRs wants an orchestration process that handles orders for this equipment. The orchestration order is broken into several fulfillment lines for each of the following: DVR, remote control, instruction manual, and extended warranty. The extended warranty is a contract purchased online, but it is not a shippable item. Therefore, it should not be sent to the fulfillment system during the Shipment task.

Note

The following is an example of a simple rule, which is well suited for rules for an orchestration process with a single line. If you want to write a rule for an orchestration process that has multiple lines, then use advanced mode rules. For more information, see Oracle Fusion Middleware User's Guide for Oracle Business Rules.

Summary of the Requirements

Create the rule while defining the SetUpShipment step. To create the rule, you must construct If and Then statements.

1. Create the If statement: If the item is shippable.
2. Create the Then statement: Then select the fulfillment line.

Creating the If Statement

1. In the **Line-Selection Criteria** column of the Manage Orchestration Process Definition page, select **Click for Rule** for the step that you are defining. Disregard all the other information above the line. Do not change the rule set name. This rule set will contain all the line selection rules you write for this step.
2. Click the **New Rule** icon.
3. You may enter a rule name. This is optional.
4. Click the arrows to expand the rule.
5. Click the **Left Value** icon to search for the left value.
6. Expand DOOFLine.
7. Select ShippableFlag.
8. Select Is.
9. In the next blank field to the right, enter "Y". You must surround the text with quotation marks.

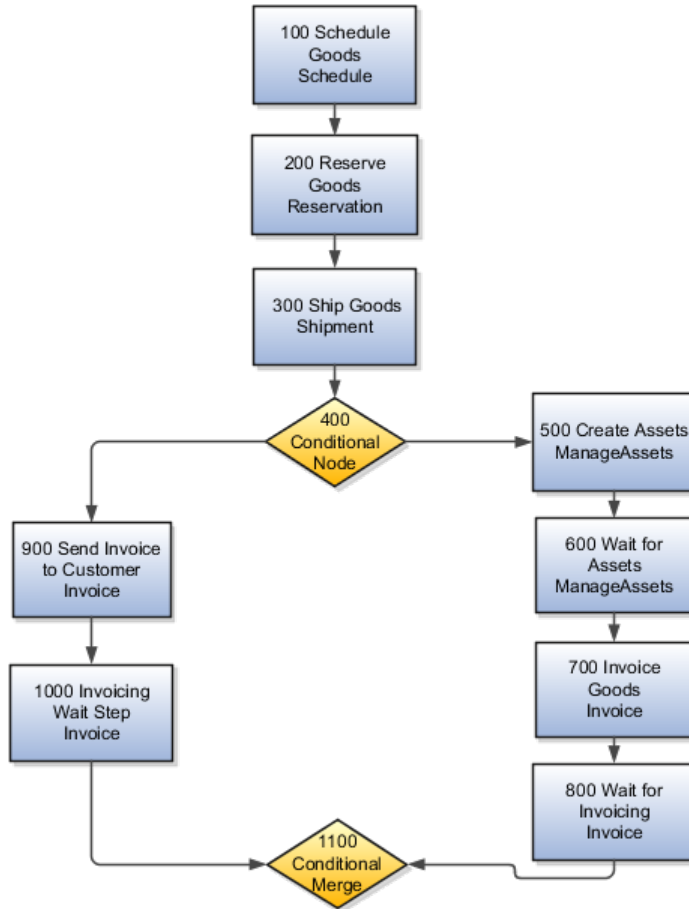
Creating the Then Statement

1. Click **Insert Action**.
2. Select Assert New. You must select this option for all line selection rules.
3. Select Result.
4. Click the **Edit Properties** icon.
5. In the **Value** field for ResultObjKey, search for DOOFLine.
6. Select FulfillLineId.
7. Click **OK**.

Branching: Explained

Use branching to create a sequence of steps that are executed only under certain conditions. A branch contains one or more steps. For example, your company sells laptop computers. If a customer buys a service agreement with the laptop computer, then you create an asset, so that the computer can be tracked. If a service agreement is not purchased, then the customer is invoiced without creating an asset.

The following figure shows an orchestration process flow that models this scenario. Each step contains the step number, task name, and task type. This example includes the ManageAssets custom task type. The conditional node indicates that an orchestration process is about to branch. The first step of the branch contains the condition. If the condition is met, then the application executes the steps on the branch that includes the Create Asset and Wait for Asset steps. Otherwise, the other branch is executed, and an invoice is created without creating an asset.



You do not need to set an **Otherwise** condition in the orchestration process definition if you have only one branch. When the orchestration process artifacts are generated, an empty default branch is added.

Creating Branching Condition Rules: Worked Example

This example demonstrates how to create a branching condition that determines whether to branch from the parent process to execute a branch. In this scenario, the order administrator of a flooring company wants an orchestration process for carpet orders. The company has a policy stipulating that a representative call a customer before sending an invoice over \$50,000.00.

Note

The following is an example of a simple rule, which is well suited for rules for an orchestration process with a single line. If you want to write a rule for an orchestration process that has multiple lines, then use advanced mode rules. For more information, see Oracle Fusion Middleware User's Guide for Oracle Business Rules.

Summary of the Requirements

Create a rule on the invoicing step of the orchestration process definition. To create the rule, you must construct If and Then statements.

1. Create the If statement: If invoice is greater than \$50,000.00
2. Create the Then statement: Then execute the branch.

This example assumes that an orchestration process is created that contains the steps necessary to carry out the fulfillment of a customer's order for carpet. This example begins with a Call Customer step. Ensure that the Call Customer step is the step after the conditional step.

Creating the If Statement

Create the If statement: If invoice (price) is greater than \$50,000.00.

1. On the Create Orchestration Process Definitions page, go to the invoicing step.
2. In the **Branching Condition** column, select **Click for Rule**.
3. Do not change the rule set name.
4. Click the **New Rule** icon.
5. Optionally, enter a rule name.
6. Click the arrows to expand the rule.
7. Click the **Left Value** icon to search for the left value.
8. Expand DOOFLine.
9. Select ExtendedAmount.
10. Click **OK**.
11. Select Is Greater.
12. In the field to the right, enter 50,000.

Creating the Then Statement

Create the Then statement: Then execute the branch.

1. Click **Insert Action**.
2. Select Assert New.
3. Select Result.
4. Select the **Edit Properties** icon.
5. In the **Value** field for ResultObjKey, search for Boolean.
6. Select True.
7. Click **Save**.

Lead Time: Explained

Lead time is the expected amount of time needed to complete a step, including wait steps and pause steps. Lead time is used to plan the orchestration process and predict expected completion dates. When real completion dates are available, they are used instead of the estimates in the orchestration process definition. The planned orchestration process appears in the Gantt chart in

the Order Orchestration work area. Lead time is also used during jeopardy calculation where jeopardy is determined by considering the number of days past lead time a step is taking.

Creating Lead-Time Rules: Worked Example

This example demonstrates how to create a lead-time rule that determines lead time for a step based on a set of conditions. The order administrator of a flooring company wants an orchestration process that handles carpet orders. The lead time for shipping the carpet is two days if the inventory organization is Houston and four days for any other inventory organization.

Note

Often, if you write a rule for an orchestration process that has multiple lines, then you should use advanced mode rules. In the following example, however, all the lines are being treated the same way, so an advanced mode rule is not required.

Creating Lead-Time Rules

In this example, you create the rule while defining the Shipment step Create Shipment. Ensure that the unit of measure is days. You must create two rules, one for when the inventory organization ID is Houston, and the other for an inventory organization ID with any other value.

Note

The Shipment task has a wait step, where a lead time can be defined, too. The lead time for the task is the sum of the lead times defined for each of the steps within the task. In this example, lead time is defined only on the Create Shipment step.

1. Create the If statement for the first rule: If inventory organization ID is 1234440.
2. Create the Then statement for the first rule: Then the lead time is equal to 2.
3. Create the If statement for the second rule: If inventory organization ID isn't 1234440.
4. Create the Then statement for the second rule: Then the lead time is equal to 4.

Creating the If Statement for the First Rule

1. In the **Lead-Time Expression** column of the Manage Orchestration Process Definition page, select **Click for Rule** for the step that you are defining. Disregard all the other information above the line. Do not change the rule set name. This rule set contains all the lead-time rules that you write for this step.
2. Click the **New Rule** icon.
3. Optionally, enter a rule name.

4. Click the arrows to expand the rule.
5. Click the **Left Value** icon to search for the left value.
6. Expand DOOFLine.
7. Select InventoryOrganizationId.
8. Click **OK**.
9. Select Is.
10. Enter 1234440.

Creating the Then Statement for the First Rule

1. Click **Insert Action**.
2. Select Assert New. You must select this option for all lead-time rules.
3. Select Result.
4. Select the **Edit Properties** icon.
5. In the **Value** column for ResultObjKey, enter 2.

Creating the Second Rule

In the same window, repeat the steps above to create a rule for the following If and Then statements. Start by clicking the **New Rule** icon. Substitute "isn't" for "is" in the first statement, and substitute 4 for 2 in the second statement.

1. If the inventory organization ID isn't 1234440.
2. Then the lead time is equal to 4.
3. Click **OK**.

Orchestration Process Planning: How It Works

You can create customized processes to manage each stage of order processing after the order is released from the order capture system. These orchestration processes include automated planning. Process planning sets user expectation of completion date of each step, task, and the orchestration process itself.

Settings That Affect Orchestration Process Planning

If you select **Replan Instantly** for an orchestration process, then the planning engine is called and the data is refreshed after each step is completed. For performance reasons, you might not want automatic replanning of some processes, especially where the step definition sequence is long or complex. Consider using this option for orchestration processes that are for high priority customer orders, or with jeopardy thresholds of less than a day. If you do not select **Replan Instantly**, then the planning data is refreshed during its normal, scheduled replan.

The following attributes affect step-specific planning:

- **Planning Default Branch:** Identifies the default path for planning. This attribute is used only if the orchestration process has conditional branches.

- **Fulfillment Completion Step:** Identifies the step at which the customer considers the lines fulfilled. This attribute is used in planning calculations to satisfy the customer request date. It is not necessarily the last step in the process definition. The chronological last step may not be last step that the customer cares about. The orchestration process is planned with the customer request date as the completion date for the step identified in the process as the last step.
- **Default Lead Time/Lead Time UOM:** Lead time is the expected duration for a given unit of work to be completed. If a lead-time expression is not defined for a step, then the default lead time is used.
- **Lead-Time Expression:** Define lead times using Oracle Business Rules. This method provides flexibility when defining complex lead-time expressions.

How Orchestration Process Planning Is Calculated

When an order enters Oracle Fusion Distributed Order Orchestration it is transformed into fulfillment lines. Then orchestration processes are created and assigned to the fulfillment lines. The orchestration process is first planned when the orchestration process is created. Planning is based on the requested date of the sales order. The requested date becomes the required completion date for the last step (step identified by the Last Fulfillment Completion Step indicator and not the chronological last step) of the orchestration process. The application then calculates the planned dates for each step and task, starting from the first chronological step, using the lead time you define. The schedule appears in the Order Orchestration work area.

The orchestration process is replanned every time an update is received from the fulfillment system. You can control when process planning occurs by scheduling a regular planning update for the frequency you want using a scheduled process.

Creating Compensation Patterns: Worked Example

This example demonstrates how to create a compensation pattern that determines what adjustments to make for a processing task in response to a requested change. The order administrator of a flooring company wants a rule that indicates that if the requested ship date is 11/20/2010, then cancel and redo the ShipGoods task.

Note

The following is an example of a simple rule, which is well suited for rules for an orchestration process with a single line. If you want to write a rule for an orchestration process that has multiple lines, then use advanced mode rules. For more information, see Oracle Fusion Middleware User's Guide for Oracle Business Rules.

Summary of the Requirements

To create the rule, you must construct If and Then statements.

1. Create the If statement: If the RequestShipDate is 11/20/2010.

2. Create the Then statement: Then redo the ShipGoods task.

Creating the If Statement

Create the If statement: If the RequestShipDate is 11/20/2010.

1. In the **Compensation Rule** column of the Manage Orchestration Process Definition page, select **Click for Rule**.
2. Do not change the rule set name.
3. Click the **New Rule** icon.
4. Optionally, enter a rule name.
5. Click the arrows to expand the rule.
6. Click the **Left Value** icon to search for the left value.
7. Expand FulfillLineTLVO.
8. Select RequestShipDate.
9. Click **OK**.
10. Select Is.
11. Click the **Right Value** icon to search for the right value.
12. Expand CurrentDate.
13. Select Date.
14. Click **OK**.

Creating the Then Statement

Create the Then statement: Then redo the ShipGoods task.

1. Click **Insert Action**.
2. Select Assert New.
3. Select Result.
4. Select the **Edit Properties** icon.
5. In the **Value** column of resultObjKey, enter "Redo". You must surround the text with quotation marks.
6. Click **OK**.

Orchestration Process Status Definition: Points to Consider

When you define an orchestration process, you must select an orchestration process class, which provides a defined set of statuses for any orchestration process to which it is applied. Use orchestration process-specific statuses to apply different sets of statuses and rule logic for different items to show the progression of the process. For example, you could have a set of statuses and rule logic for orchestration processes for textbooks for customers that are colleges and a different set of statuses and rule logic for orchestration processes for textbooks for customers that are primary schools.

If you choose not to customize the status condition for an orchestration process, then the default statuses are used. If you customized the name of the default status, then the status appears in the application.

Orchestration Process Classes

The orchestration process class is a set of status codes. When you select a process class in the header, the status codes from that class are available for selection when you create the status conditions. These are the status codes that will represent the status of the orchestration process and will be seen throughout the application. The status code is also used for grouping orchestration processes by status to enable order managers to quickly identify orchestration processes that are in the same status.

Orchestration Process Status Rules

You can set up rules that govern under what conditions a status is assigned to an orchestration process. For example, you could create a rule that says if the status of the Schedule task is Not Started, then assign the orchestration process the status Unscheduled. You must designate a status or set of statuses to indicate that a task is complete. You can only select from those that were defined to mark a task complete.

Orchestration Process Status: How It Is Determined

During processing of an orchestration order, the tasks of the assigned orchestration process are fulfilled step by step. A default set of sequential statuses is provided for the fulfillment tasks, but you can also create your own fulfillment task statuses and sequences for an orchestration process. You must determine the status that will be assigned to an orchestration process at each stage of the process. For example, if a Schedule Carpet task has a status of Unsourced, what status should the orchestration process have?

Settings That Affect Orchestration Process Status

You can designate the statuses that represent an orchestration process when you define the orchestration process. These statuses are used to represent the status of the orchestration process throughout the application. You can select a preset group of orchestration process statuses by selecting an orchestration process class. You can create rules that govern how statuses are attained during the orchestration process and the conditions under which they are assigned to the orchestration process task.

How Orchestration Process Status Is Determined

If rules are created, then at run time the application evaluates each of the statements sequentially. The true condition with the highest sequence number determines the status of the orchestration process.

When a fulfillment line splits, the resulting fulfillment lines have duplicate tasks. At some point, the tasks could have different statuses. For example, the Schedule task for fulfillment line A1 is in status Not Scheduled, and the Schedule task

for fulfillment line A2 is Scheduled. In this case, the application searches for the split priority of the task statuses. The status with the higher split priority (lower number) becomes the status of the orchestration process.

Creating Orchestration Process Status Conditions: Worked Example

This example demonstrates how to create status conditions for an orchestration process. A company that sells flooring needs an orchestration process that reflects the steps required to fulfill such orders. The orchestration process definition must designate how to reflect the status of the orchestration process at any point in time. The status of the orchestration process is based on the status of the tasks. This example shows how to create the conditions that designate the status of the orchestration process.

When you create an orchestration process status condition, you must decide which orchestration process class to use and which statuses you want to reflect the status of the orchestration process. An orchestration process class is a set of statuses that can be used for an orchestration process.

Prerequisites

This example assumes that an administrator has created an orchestration process class called Carpet Class on the Manage Status Values page.

1. On the Manage Status Values page, create an orchestration process class called Carpet Class.
2. For the Schedule task type, include the following statuses: Started, Canceled, Not Started.

Summary of the Requirements

1. Create an orchestration process definition called Standard Carpet.
2. Select the Carpet Class orchestration process class.
3. Create the orchestration process status condition: If the ScheduleGoods task status is Scheduled, then use Scheduled to represent the status of the orchestration process.

Creating an Orchestration Process Status Condition

This example shows you how to create one orchestration process status condition. Repeat these steps for all the status conditions you need. You are not required to use all the statuses in the orchestration process class.

1. On the Manage Orchestration Process Definitions page, click **Create**.
2. In the **Process Name** field, enter Standard Carpet.
3. In the **Process Class** list, select Carpet Class.
4. On the Step Definition tab, select **Add Row**.
5. In the **Step Name** field, enter ScheduleGoods.
6. In the **Step Type** field, select Service.
7. In the **Task Type** field, select Schedule.

8. In the **Task Name** field, select ScheduleGoods.
9. In the **Service Name** field, select Create Scheduling.
10. Click **Save**.
11. Select the Status Conditions tab.
12. On the Orchestration Process Status Values tab, select **Add Row**.
13. In the **Sequence** field, insert 1.
14. In the **Status Value** list, select Scheduled to represent the first status of the orchestration process: .
15. Click the **Expression** icon.
16. On the Tasks tab, select ScheduleGoods.
17. Click **Insert Into Expression**.
18. On the Operators tab, select =.
19. Click **Insert Into Expression**.
20. On the Tasks tab, expand Schedule.
21. Select Scheduled.
22. Click **Insert Into Expression**.
23. Click **OK**.

Fulfillment Line Status Definition: Points to Consider

When you create an orchestration process definition, you can opt to define status conditions for certain types of fulfillment lines that can be processed by the orchestration process. Use fulfillment line-specific status conditions to apply different sets of statuses and rule logic for different items. For example, you could have one set of status conditions for textbooks and another set for paperback books.

If you choose not to create status conditions for a fulfillment line, then the status conditions with the status rule set that is assigned to the default category dictates the status progression.

Status Catalogs and Categories

Your organization might need for different fulfillment lines within the same orchestration process to have different status progressions. For example, a model with shippable fulfillment lines and nonshippable fulfillment lines may require different statuses for each type of fulfillment line. A status catalog provides a means to group like items, so that they can achieve the same statuses at the same time. Status catalogs are defined in Oracle Fusion Product Model.

You can select a status catalog when you create an orchestration process definition. Status catalogs that meet the following criteria are available for selection:

- An item exists in only one category within a catalog.

- A category contains items or subcategories but not both.
- A catalog is controlled at the master level only, not at the organization level.

You can use catalogs and categories in multiple orchestration process definitions. Use a category to ensure that the same set of status conditions is applied to specific sets of fulfillment lines. The same status conditions are applied to all fulfillment lines that have the item that belongs to that category.

Status Rule Set

Whether or not you use status catalogs, you can use status rule sets to apply a set of sequential statuses to the fulfillment line that is processed by the orchestration process. A status rule set is a set of rules that govern the conditions under which status values are assigned to fulfillment lines. When you create a status rule set, you determine the status that will be assigned to a fulfillment line at each stage of the process. For example, if an item has a status of Unsourced, then the fulfillment line will have the status Unscheduled. A status rule set streamlines administration by enabling you to use a rule set with any number of fulfillment lines, rather than by entering separate rules for each fulfillment line. You can also apply the same logic to multiple categories.

In the case where a parent and a child category refer to different status rule sets, the child takes priority. This allows you to define an All category to handle all items in one definition, as well as to add an additional subcategory for a subset of products that needs to use a different status rule set.

During order processing, the application assigns an overall status to each orchestration order. This status is determined by assigning the orchestration order the status of the fulfillment line that has progressed the furthest in the order life cycle. To determine the fulfillment line status, the application evaluates each of the status conditions of the fulfillment line sequentially. The true condition with the highest sequence number determines the status of the fulfillment line.

Caution

If you used the Functional Setup Manager migration tool to port test instance data to a production environment, then do not change the status rule set name in either instance. Changing the name could prevent references to other data in the orchestration process from being updated.

Creating Fulfillment Line Status Conditions: Worked Example

This example demonstrates how to create status conditions for a fulfillment line with several items that require different statuses. A flooring company is setting up orchestration processes to handle orders for different types of flooring. The same orchestration process could be used for multiple types of flooring, but the administrator wants to define statuses for each type of flooring separately because they require slightly different statuses. This example demonstrates how to select the status catalog and create the status conditions for a single category of items within an orchestration process.

When you create an orchestration process, you need to decide whether you want different fulfillment lines that get assigned to the process to have different statuses as they progress through fulfillment. If so, you must determine how to group the fulfillment lines using catalogs and categories.

The Flooring catalog has the following categories: Carpet, Tile, Hardwood. You select the category for Carpet. You create a status rule set with conditions that will yield the following statuses: Not Scheduled, Scheduled, Awaiting Shipment, Shipped, Billed.

Prerequisites

This example assumes that an administrator has created fulfillment line status values on the Manage Status Values page. This example also assumes a Flooring catalog was created in Oracle Fusion Product Model.

1. On the Manage Status Values page, create fulfillment line status values.
2. In Product Model, create a Flooring catalog with the following categories: Carpet, Tile, Hardwood.
3. Assign the carpet items to the carpet category.

Summary of the Requirements

1. Create an orchestration process definition called Standard Flooring.
2. Select Flooring as the status catalog.
3. Create the status rule set Carpet Rule Set, and assign it to the Carpet category within the Standard Flooring orchestration process.
4. Create the following fulfillment line status condition for Carpet Rule Set: If the Schedule task status is Started, then use Scheduled to reflect the status of the fulfillment line.

Creating a Fulfillment Line Status Condition

This example shows you how to create one fulfillment line status condition. Repeat these steps for all the status conditions you need.

1. On the Manage Orchestration Process Definitions page, click **Create**.
2. In the **Process Name** field, enter Standard Flooring.
3. In the orchestration process definition header, select the Flooring status catalog.
4. On the Step Definition tab, select **Add Row**.
5. In the **Step Name** field, enter ScheduleGoods.
6. In the **Step Type** field, select Service.
7. In the **Task Type** field, select Schedule.
8. In the **Service Name** field, select Create Scheduling.
9. Click **Save**.
10. In the Fulfillment Line Status Values tab, select and add the Carpet status category.
11. In the **Status Rule Set** column, click the arrow and select Create.

12. In the field type Carpet Status Rule Set, ensure that Create New is selected, and click **Save and Close**.
13. Click **Save** at the top of the page.
14. Click **Edit Status Rule Set**.
15. Add a row.
16. On the Tasks tab, select ScheduleGoods.
17. Click **Insert Into Expression**.
18. On the Operators tab, select =.
19. Click **Insert Into Expression**.
20. On the Tasks tab, expand Schedule.
21. Select Scheduled.
22. Click **Insert Into Expression**.
23. Click **OK**.
24. In the Status Value column, select Scheduled.
25. Click **OK**.

Orchestration Process Definition Deployment: Explained

After you finish creating or updating an orchestration process definition, you must release it and then deploy it on an instance of Oracle Fusion Distributed Order Orchestration. Deploying the orchestration process makes it available for use by the application.

If you want to use the ShipOrderGenericProcess and ReturnOrderGenericProcess predefined orchestration processes, then you must generate them and deploy them; it is not necessary to release them.

After your orchestration process is defined, you must take the following steps to deploy it:

- Release the orchestration process definition.
- Download the orchestration process definition.
- Modify the SOA configuration plan.
- Deploy the JAR file using the modified configuration plan.

Do not modify orchestration process definitions outside of the Manage Orchestration Process Definition pages.

Release the Orchestration Process Definition

When an orchestration process is released it is automatically validated. After you release the orchestration process definition, batch-level validations are performed to ensure that the orchestration process was constructed correctly. If any errors are generated during validation, the release process stops and an error icon appears next to the orchestration process name. The list of errors is retained until the next time the batch validation runs. If the orchestration process is valid, then release of the process continues. An orchestration process is valid

if no error messages are generated; warning messages may be associated with a valid process. After validation is complete, the orchestration process definition becomes read-only. At this point, the orchestration process is given Released status, and the BPEL artifacts needed to deploy and run the orchestration process are created and stored.

Download the Orchestration Process Definition

After you release an orchestration process definition, you deploy the downloaded artifacts to the server. Use Oracle Fusion Setup Manager to export the artifacts. Oracle Fusion Middleware is used to deploy artifacts.

1. On the Manage Orchestration Process Definitions page, select the orchestration process that you want to deploy.
2. Click the **Edit** icon.
3. In the Download Generated Process window, click **Download**.
4. Save the archive file that appears to a local directory.
5. Open the archive file in a local directory.

The JAR file is located in a Deploy folder within a folder bearing the name of the orchestration process that you downloaded.

Modify the SOA Configuration Plan

Modify the SOA configuration plan, replacing the host names and ports with your organization's Distributed Order Orchestration ADF server and port and Distributed Order Orchestration (Supply Chain Management) SOA server and port. Use the external-facing URLs of the servers. The configuration plan enables you to define the URL and property values to use in different environments. During process deployment, the configuration plan is used to search the SOA project for values that must be replaced to adapt the project to the next target environment.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAConfigPlan
xmlns:jca="http://platform.integration.oracle/blocks/adapter/fw/metadata"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:orawsp="http://schemas.oracle.com/ws/2006/01/policy"
xmlns="http://schemas.oracle.com/soa/configplan">
<composite name="*">
<import>
<searchReplace>
<search/>
<replace/>
</searchReplace>
</import>
<service name="client">
<binding type="ws">
<attribute name="port">
</attribute>
</binding>
</service>
<reference name="*">
<binding type="ws">
<attribute name="location">
<searchReplace>
<search>http://localhost_am:port</search>
<replace>http://actualD00ADFserver:port</replace>
</searchReplace>
```

```
<searchReplace>
<search>http://localhost_soa:port</search>
<replace>http://actualDOOSOAServer:port</replace>
</searchReplace>
</attribute>
</binding>
</reference>
</composite>
</SOAConfigPlan>
```

Deploy the JAR File

To deploy the JAR file, you can use any of the following: Oracle Enterprise Manager Fusion Middleware Control, ant command line tool, or Oracle WebLogic Scripting Tool. For more information about deploying SOA composite applications, see Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite.

Process Assignment Rules: Explained

After Oracle Fusion Distributed Order Orchestration creates an orchestration order, the application assigns orchestration processes to fulfillment lines based on process assignment rules. Process assignment rules are executed in the Oracle Business Rules engine. Process assignment rules are built based on orchestration groups and using orchestration order attributes.

You do not need to specify versions or effectivity dates in the process assignment rules because versions and effectivity dates are controlled at the orchestration process level.

Orchestration Groups

A fulfillment line belongs to an orchestration group. Distributed Order Orchestration contains the following predefined orchestration groups: Shipment Set, Model/Kit, and Standard. Standard is used for standard items or finished items. All the fulfillment lines that belong to a shipment set or a model are assigned the same orchestration process.

Assign a process for each set of unique conditions. You can set up a default orchestration process for each orchestration group using the Otherwise construct.

Before you create process assignment rules, you must define orchestration processes or at least know the names you will give to orchestration processes. You will add the orchestration process names to bucket sets to make them available for selection when you create a process assignment rule.

Creating Process Assignment Rules: Examples

Use these scenarios to understand how to use process assignment rules.

Assigning an Orchestration Process According to Product

All orders for ceramic tile must undergo the same processing steps, so you write a process assignment rule that assigns the Tile Processing orchestration process to all orchestration order lines with tile as the product.

Assigning an Orchestration Process According to Customer

Customer A requires an extra inspection step for all its orders, so you write a process assignment rule that assigns the Customer A Process to all orchestration order lines that have Customer A in the orchestration order header.

Assigning an Orchestration Process According to Ship-to Region

Orders that are bound for countries outside your current location require different handling, such as completion of customs forms. You write a process assignment rule that assigns the International Orders orchestration process to all orchestration order lines that have a foreign country in the ship-to address in the header.

Creating an Orchestration Process Assignment Rule: Worked Example

This example demonstrates how to create an orchestration process assignment rule that assigns a specific orchestration process to large orders.

Summary of the Tasks

The example shows how to write a process assignment rule that assigns an orchestration process based on the ordered quantity.

Rules consist of conditions and actions. If the conditions are true, then the actions are taken. In this example, you will create a rule that says: If ordered quantity is large, then assign the LargeOrderProcess orchestration process to the fulfillment line. You will:

1. Create a decision table.
2. Add a new condition for ordered quantity to the decision table. This is the If clause. In this activity, the If clause is: If ordered quantity is Large...
3. Create a bucket set that contains the quantity ranges for comparison to the runtime value.
4. Use the bucket set in the condition that you created.
5. Select an action. This is the Then clause. In this example, the Then clause is: Then assign the orchestration process "LargeOrderProcess" to this fulfillment line.

Navigating to the Manage Orchestration Process Assignment Rules Page

1. On your implementation project, expand the Define Sales Order Fulfillment task list.
2. Expand the Define Orchestration task list.
3. Click the Go to Task icon for the Manage Orchestration Process Assignment Rules task.

Creating a Decision Table

1. Next to the View list, click the arrow next to the Add icon and select Add Decision Table. This action displays a decision table with one rule (R1). Rules are read vertically. You will be adding information to this decision table.

2. Substitute Decision Table 1 with LargeOrdersRule.
3. Click the **Save** button.

Adding a New Condition to the Decision Table

1. In the View list, select the decision table that you just created.
2. To the left of the decision table name, click the Show Advanced Settings icon.
3. Select the **Advanced Mode** check box. If other check boxes are also selected, then you may leave them as they are. You are using advanced mode to create aliases and declare instances for some order orchestration variables. Creating aliases can make it faster to use variables when writing a rule. In this case, these particular variables are also aliased in the predefined decision table AssignProcess. The same aliases have been used in this example, but this is not required because they are local references, not global references.
4. Click Collapse Pane to show more of the screen, if needed.
5. In the field that appears to the left of "is a," enter Group.
6. In the list that appears to the right of "is a," select AssignLaunchRules.ResponseRL. You now have an alias for the AssignLaunchRules.ResponseRL. This means you can refer to AssignLaunchRules.ResponseRL as Group.
7. Click the Add Pattern icon.
8. In the field that appears to the left of "is a," enter: fline.
9. In the list that appears to the right of "is a," select AssignLaunchRules.FulfillLineVO. You are creating an alias for AssignLaunchRules.FulfillLineVO. You can now refer to AssignLaunchRules as fline.
10. Directly above the decision table, click the arrow next to the Add icon and select Add Condition.
11. The Condition Browser appears. Notice that the variables that you declared appear as the aliased names.
12. Expand fline.
13. Select OrderedQty, and click the **OK** button. You have just created part of the If clause: If ordered quantity is equal to X. Next you will create the quantity buckets that you can select to complete the If clause.

Creating a Bucket Set List

1. Click the cell where you created the condition.
2. In the Select Bucketset list, select the Local List of Ranges list item to create a set of values to be compared to the ordered quantity at runtime.
3. Click the Edit Bucketset icon next to Local List of Ranges.
4. In the **Name** field of the Bucketset Editor window, enter a valid name, such as Bulk Ranges.
5. In the **Description** field, enter a valid description, such as the following:
Set of ranges to use for bulk fulfillment.

6. In the Data Type list, select the int list item. Only data types that match the previously selected condition attribute are displayed.
7. Click the Add Bucket icon.
8. In the **Description** field for the end point of 1, enter a valid description, such as Small.
9. Click the Add Bucket icon.
10. In the **End Point** field, change the value to 100.
11. In the **Description** field, enter a valid description, such as Medium.
12. Click the Add Bucket icon.
13. In the **End Point** field, change the value to 1000.
14. In the **Description** field, enter a valid description, such as Large. For this exercise we created three buckets, but you could create more or fewer.
15. Click the **OK** button.

Using the Bucket Sets in the Condition

1. Double-click in the rule (R1) next to the condition that you created. The cell contains a question mark. A dropdown list appears.
2. In the list select ≥ 1000 . Click elsewhere on the page to close the list.

Configuring the Action to Take if the Condition Is True in the New Rule

1. Click the arrow next to the Add icon, and select Add Action and then Modify. The Action Editor appears.
2. In the Target box, select Group. The attributes associated with the view object appear in the Arguments box.
3. Select the check box in the Parameterized column for processName. That indicates that you want to fill in the value separately for each rule.
4. Click the **OK** button. You are returned to the decision table.
5. Double-click the cell directly to the right of processName:String.
6. Enter "LargeOrderProcess". Quotation marks are required.
7. Select the check box on the line above the cell from the previous step. Selecting the check box means that you want to use this action for this rule.
8. Click the **Save** button.

Define Processing Constraints

Define Processing Constraints: Overview

Each company has its own business rules, which must be applied during the orchestration process. The constraint framework allows for the implementation of those specific requirements. Processing constraints are rules that control attempted changes to an order: What can be changed, when, and by whom.

At runtime, processing constraints are checked on changes to orchestration orders, orchestration order lines, and fulfillment lines. Changes that are not permitted by processing constraints are not allowed. A message is returned indicating the reason the change is not permitted.

Processing constraints are used also to validate the required attributes for fulfillment requests.

Some processing constraints are predefined; you cannot change these processing constraints. If you want to make processing constraints more restrictive, then you must create new ones.

Using Processing Constraints: Examples

Consider using processing constraint in scenarios such as the following. In all of these scenarios, the change is submitted, but it is never processed because the processing constraint rejects it. A message is returned to the order capture application indicating that the change could not be made because of the processing constraint.

Constraint Prohibits Changes at Shipping Stage

An orchestration process gets to the shipping stage. Then a change order is submitted against the orchestration order. The orchestration process is so far along that it is costly and impractical to make changes. To prevent this problem, you create a processing constraint that rejects any changes when an orchestration process is in the shipment step.

Constraint Rejects Orders Without a Required Attribute

Your company has a policy that it does not deliver items to an address that does not have a ship-to contact. Sometimes sales orders that do not have a ship-to contact are submitted. To prevent this problem, you create a processing constraint that rejects sales orders that do not have the required information.

Constraint Prohibits Changes by Unauthorized User

Your company allows customer service representatives to submit certain customer changes without approval from a manager. If the change order has a transaction value over \$100, then the change must be submitted by a manager. You create a processing constraint that rejects changes orders with transaction values over \$100 from anyone with the customer service representative role.

Creating Processing Constraints: Worked Example

This example demonstrates how to create a processing constraint that prevents changes to any orchestration process that is in the shipping phase. An orchestration process gets to the shipping stage. Then a change order is submitted against the orchestration order. The orchestration process is so far along that it is costly and impractical to make changes. To prevent this problem, you create a processing constraint that rejects any changes when an orchestration process is in the SetUpShipment step.

Before you create a processing constraint, you must create a constraint entity, a validation rule set, and a record set.

Summary of Requirements

1. Create the Shipping service as a constraint entity.
2. Create a validation rule set.
3. Create a record set.
4. Create a processing constraint that prohibits changes when the orchestration process reaches the shipping stage.

Creating a Constraint Entity

This is the process of creating a process task entity so that it can be used later to create a processing constraint. This is the entity that will be constrained.

1. On the Manage Constraint Entities page, click the **Create** icon.
2. In the **Entity Name** field, enter Shipping Service.
3. In the **Description** field, enter the following: Shipping service entity.
4. From the list of services, select the Create Shipping service; do not select a process name or task name.
5. Click **Save**.

Creating a Validation Rule Set

A validation rule set names a condition and defines the semantics of how to validate that condition for a processing constraint.

1. On the Manage Processing Constraints page, select the Validation Rule Set tab.
2. Click the **Add Row** icon.
3. In the **Name** field, enter Shipment VRS.
4. From the **Validation Type** list, select Process.
5. Select the Create Shipping service
6. Click **Save**.

Creating a Record Set

A record set is a group of records that are bound by common attribute values. You can define conditions and specify a record set to be validated for a given condition as defined by the validation template.

1. On the Manage Processing Constraints page, select the Record Set tab.
2. Click the **Add Row** icon.
3. In the **Name** field, enter Fulfillment Lines That Belong to Same Customer.
4. In the **Description** field, enter the following: A record set created on fulfillment lines that belong to the same customer.
5. In the **Short Name** field, enter FFLCUST.

6. In the **Entity** list, select Order Fulfillment Line.
7. In the Attributes region, click the **Add Row** icon.
8. Select Bill-to customer ID.
9. Click **Save**.
10. Click Generate Packages.

Creating a Processing Constraint

Now that you have created a constraint entity, validation rule set, and record set, you can create the processing constraint.

1. On the Manage Processing Constraints page, select the Constraints tab.
2. Click the **Add Row** icon.
3. In the **Constraint Name** field, enter Shipping Constraint.
4. From the **Constraint Entity** list, select Order Fulfillment Line.
5. From the **Constrained Operation** list, select Update.
6. On the Conditions tab, click the **Add Row** icon.
7. In the Group Number field, enter 10.
8. From the **Validation Entity** list, select Order Fulfillment Line.
9. From the **Validation Rule Set** list, select Shipment VRS.
10. From the **Record Set** list, select Fulfillment Lines That Belong to Same Customer.
11. Create the following message: The fulfillment line could not be updated because it is in the shipping phase.
12. On the Applicable Roles tab, select **All Roles**.
13. Click **Save**.

Define Transformation Details

Sales Orders: How They Are Transformed to Orchestration Orders

When sales orders enter Oracle Fusion Distributed Order Orchestration from disparate order capture applications, they must be transformed into business objects that can be processed by Distributed Order Orchestration. During this process, called decomposition, sales orders are deconstructed and then transformed into Distributed Order Orchestration business objects.

Settings That Affect Sales Order Transformation

Business rules determine how sales orders are transformed. The following business rules are available: Pretransformation defaulting rules, product transformation rules, posttransformation defaulting rules, and process assignment rules.

How Sales Orders Are Transformed

Sales orders are transformed as follows:

1. The sales order is passed from the order capture application.
2. The connector service transforms the sales order from an external order capture system to a canonical business object called the sales order enterprise business object. The sales order enterprise business object structurally transforms the sales order from an external order capture system to an orchestration order in Distributed Order Orchestration. The Receive and Transform service, SalesOrderOrchestrationEBS, looks up the cross-reference values from the customer data hub, Oracle Fusion Trading Community Model, and Oracle Fusion Product Model to determine whether the sales order values must be transformed to common values to populate the sales order enterprise business object. Cross-referencing is required for static data such as country code and currency codes, as well as for dynamic data such as customers and products. The attributes come from different sources: Product Model, Trading Community Model, and the order orchestration and planning data repository. If the order capture and Distributed Order Orchestration systems use different domain values, then the connector service transforms the structure and values from the order capture system domain to the Distributed Order Orchestration domain. The Receive and Transform service is called in the default predefined process prior to storing the sales order.
3. The connector service calls the decomposition process composite enterprise business function through a decomposition enterprise service. The decomposition process composite is exposed as a WSDL that can be called as a service from the connector service through an enterprise business service.
4. The decomposition service calls the requested operation (create, delete, update, or cancel orchestration order).
5. The decomposition service accepts the sales order enterprise business message as input. The decomposition service returns a sales order enterprise business message as the output.
6. The product is transformed according to the business rules that you write.
7. The Assign and Launch service assigns orchestration processes to line items according to the business rules that you write.

Order capture services are used to communicate updates back to the order capture system. To receive updates, the order capture system must subscribe to the events.

Connector for Sales Order Transformation: Explained

The connector service transforms the sales order business object as understood by an order capture application to an enterprise business object. The connector service then calls the Receive and Transform service.

The connector transforms the structure and content of the sales order.

Structural Transformation

The connector service transforms the sales order from an external order capture system to a canonical business object called the sales order enterprise business object. The sales order enterprise business object structurally transformations the sales order from an external order capture system to an orchestration order in Oracle Fusion Distributed Order Orchestration. The decomposition service accepts a sales order enterprise business message as the input and returns a sales order enterprise business message as the output. You can create the connector according to your organization's requirements, using the sales order enterprise business object attributes that are used by Distributed Order Orchestration.

Note

The Sales Order enterprise business object and front-end connector accept only the internal IDs of business objects.

Cross-Referencing: Explained

You must establish and maintain cross-references to relate business data between different integrated order capture and fulfillment systems and Oracle Fusion Distributed Order Orchestration.

Note the location of and other pertinent information about the following cross-references:

- Customer cross-references
- Item cross-references
- Other cross-references

Customer Cross-References

Customer cross-references are maintained in the Oracle Fusion Trading Community Model. You can use external customer hubs with Distributed Order Orchestration, but you must maintain cross-references in Trading Community Model also, so that Distributed Order Orchestration can resolve the Oracle Fusion customer values and vice versa. You can capture or set up customer cross-references in the Oracle Fusion customer model as part of the customer creation and update process.

During order processing, the order is created in the order capture system and sent to Distributed Order Orchestration, along with customer data. If the customer already exists in the Fusion customer master, then Distributed Order Orchestration uses a cross-reference to obtain the master customer record and the customer ID for the intended order fulfillment system. Then the decomposed order is sent, along with the customer ID and necessary attributes from the master.

Item Cross-References

Item cross-references are maintained in the Oracle Fusion Product Model. The cross-reference is established between the source system item and the item in the master product information repository, which is the Product Model. Two types of relationships are used for the cross-references: Source system item relationship, which captures the relationship between the source item and the Fusion item when a product hub is used; and a named item relationship, which is used to store the cross-reference between the source item and the Fusion item. This type of relationship is used when items are brought from disparate systems into a master product information repository. A hub is not used in the latter scenario.

Other Cross-References

The cross-references of all attributes, except customer and item attributes, are maintained in the order orchestration and product data repository. Use domain value maps for attributes from the order orchestration and planning data repository. Domain value maps are used during the collections process to map the values from one domain to another, which is useful when different domains represent the same data in different ways.

Product Transformation Setup: Explained

Set up product transformation to ensure that the products are converted properly when a sales order is transformed into an orchestration order.

Product transformation is executed using a combination of product relationships, product structures, transactional item attributes, and business rules. Product transformation setup consists of the following steps

1. Define items in Oracle Fusion Product Model.
2. Define rule bucket sets.
3. Set the bucket sets to the facts.
4. Create rules.

Define Items in Product Model

You must set up items and their structures and attributes in Product Model and then map them to fulfillment products.

- Define products used for product transformation rules in the product master.
- If you plan to base transformation rules on product structure, then define product structures.
- If you plan to use attribute-based rules, then define transactional item attributes.
- Define the relationship between sales product and fulfillment product.

Define Rule Bucket Sets

Create bucket sets on the Managing Product Transformation Rules page. Bucket sets contain the options that are available for selection when creating rules. Smaller bucket sets are more likely to be reused.

Create Rules

Create product transformation rules on the Managing Product Transformation Rules page.

Pretransformation Defaulting Rules: Explained

Use pretransformation defaulting rules to automatically populate specific attributes onto an orchestration order before product transformation. You can use the defaulted attribute value in the product transformation rules.

The master inventory organization automatically is defaulted to the orchestration order, so that it is available for product transformation rules.

Creating Pretransformation Defaulting Rules: Examples

Use these scenarios to understand how to use pretransformation defaulting rules.

Automatically Populating an Attribute on the Fulfillment Line

Your company receives sales orders for widgets that have an attribute called Request Date. You want this attribute to appear on all fulfillment lines for widgets. You write a pretransformation defaulting rule that states that if the product is a widget then populate the Request Date attribute on the fulfillment line.

Automatically Populating an Attribute for Use in a Product Transformation Rule

Your company receives sales orders for widgets. You want to write a product transformation rule that converts the widget size from centimeters to inches, but you must first populate the fulfillment line with the Size attribute. You write a pretransformation defaulting rule that says that if the product is a widget then populate the fulfillment line with the Size attribute.

Product Transformation Rules: Explained

During product transformation, a sales-oriented representation of the products on a sales order is transformed to a fulfillment-oriented representation of the products on an orchestration order. Product transformation is effected using a combination of product relationships, product structures, transactional item attributes, and business rules. You create transactional item attributes and

product relationships and structures in Oracle Fusion Product Model. You write rules on the Manage Product Transformation Rules page in Distributed Order Orchestration.

The following types of product transformation are supported:

- Attribute to attribute
- Attribute to product
- Product to attribute
- Product to product
- Context to attribute
- Context to product

Creating Product Transformation Rules: Examples

Use the following examples to understand the types of product transformation rules you can write.

Creating Attribute-to-Attribute Transformation Rules

Your US-based company receives sales orders from its office in Europe. The item size on the sales order line is expressed in centimeters, but you want it to appear in inches on the orchestration order line. You write an attribute-to-attribute transformation rule that transforms the transactional item attribute from the source order line to a different attribute on the orchestration order line.

Creating Attribute-to-Product Transformation Rules

Your company receives sales orders for MP3 players with various transactional item attributes, such as color and storage capacity. You want each combination of attributes to correspond to a product number, for example, an MP3 player of color silver and storage capacity of 8 megabytes would appear as MA980LL/A on the orchestration order line. You write an attribute-to-product transformation rule transforming the attributes to a product number.

Creating Context-to-Attribute Transformation Rules

Your company manufactures laptop computers. Some are shipped to domestic locations, and others are shipped to international locations. Each type of shipping has different requirements. You write a context-to-attribute transformation rule that transforms the region, or context, on the sales order line into a packing type attribute on the orchestration order line.

Creating Context-to-Product Transformation Rules

Your company receives sales orders for laptop computers from different geographical regions. The geographical region of the order determines which adapter is included with the product. You write a context-to-product transformation rule that transforms a single sales order to an orchestration order with two lines, one of which is reserved for the region-specific adapter:

- Orchestration order line 1: laptop computer
- Orchestration order line 2: 65-watt AC adapter

Creating Product-to-Product Transformation Rules

Your company receives sales orders for camcorders that come with several accessories: Lithium-ion battery, AC adapter, editing software, packing materials. You write a product-to-product transformation rule that creates five orchestration order lines:

- Orchestration order line 1: camcorder
- Orchestration order line 2: lithium-ion battery
- Orchestration order line 3: AC adapter
- Orchestration order line 4: editing software
- Orchestration order line 5: packing materials

Creating Product-to-Attribute Transformation Rules

Sales orders contain attributes for width and height, but you want an attribute for area on the orchestration order. You write a product-to-attribute transformation rule that computes the value for area from the width and height transactional item attributes and places it on the orchestration order.

Creating a Product Transformation Rule with a Bucket Set: Worked Example

This example shows how to create a product transformation rule with a bucket set in a decision table.

Create a bucket set that contains the values you will select when creating a product transformation rule that represents the following offer:

Customers who order an printer model T0115 get expedited shipping.

This example includes the following tasks:

1. Navigate to the Manage Product Transformation Rules Page
2. Create a Bucket Set List
3. Create a Decision Table
4. Add a New Condition to the Decision Table
5. Add a New Rule to the Decision Table
6. Configure the Action to Take if the Condition Is True in the New Rule

Navigating to the Manage Product Transformation Rules Page

1. On your implementation project, expand the Define Sales Order Fulfillment task list.

2. Expand the Define Transformation Details task list.
3. Click the **Go to Task** icon next to Manage Product Transformation Rules.

Creating a Bucket Set List

Oracle Business Rules uses long item IDs from the database. Users might not recognize the long IDs, so create buckets that list the item IDs and aliases.

Note

In this example, you will create two buckets but select only one.

1. Click **Bucketsets** at the top left of the page.
2. Click the arrow next to the **Add Bucketset** icon and select List of Values. A new bucket set with the placeholder text Bucket Set 1 appears.
3. Select Bucket Set 1, if it is not selected already. The **Edit Bucketset** icon appears above the table. If it does not appear, click the **Expand Pane** icon, or press F5.
4. Click the **Edit Bucketset** icon. The Bucketset Editor window appears.
5. In the **Name** field, overwrite the existing text and enter Printer IDs.
6. Enter the following description: List of printer item IDs and aliases for selection in product transformation rules.
7. In the Data Type list, select string.
8. Click the **Add Bucket** icon. A new **Value** field appears
9. In the **Value** field, replace Bucket 1 with 200000003357223L. This is the item ID from the database. L denotes a long value.
10. In the **Alias** field (which was populated with the same number as the **Value** field), overwrite the existing number with T0115. This is the item code that is on the customer catalog and is used throughout the company.
11. In the **Description** field, enter T0115 printer. It is possible to create more buckets, but you need only one for this example.
12. Click **OK**.
13. Click **Save**.

Creating a Decision Table

1. Next to the View list, click the arrow next to the **Add** icon and select Add Decision Table from the list.
2. Replace Decision Table 1 with AddExpeditedShippingforProductT0115.
3. Click **Save**. The page refreshes.
4. In the View list, select your decision table.
5. Click the **Collapse Pane** icon, if necessary, to see more of the page.

Adding a New Condition to the Decision Table

1. Scroll down to the table.

2. Directly above the table, click the arrow next to the **Add** icon, and select Add Condition. The Condition Browser window opens.
3. Expand the OrderTransformationRules list item. The new condition will be based on the item ID.
4. Expand FulfillLineVO.
5. Expand InventoryItemId.
6. Select toString.
7. Click **OK**.

Adding a New Rule to the Decision Table

1. In your decision table, click the field that shows your condition. A field with a list of values appears.
2. Click the arrow next to Local List of Values.
3. Select the bucket set that you created. Only bucket sets of the matching data type appear in the list.
4. Double-click in the R1 column for the condition. This cell has a question mark. A dropdown list appears.
5. Click the dropdown list arrow.
6. Select T0115. Click elsewhere on the page to close the selection list.

Configuring the Action to Take if the Condition Is True in the New Rule

1. Click the arrow next to the **Add** icon and select Add Action.
2. Select **Modify**. The Action Editor opens.
3. In the Target list, select OrderTransformationRules.FulfillLineVO.
4. Press the Page Down key to see more arguments.
5. Click in the Arguments list, and then press the Page Down key to see more arguments.
6. Next to shipClassOfService select the Parameterized check box. You will have to press Page Down about five times to get to shipClassOfService.
7. Click **OK**.
8. Click the **Collapse Pane** icon, if necessary, to see more of the page.
9. Double-click the cell to the right of shipClassOfService.
10. Enter "Expedited". Note that quotation marks are required.
11. Click **Save**.

Creating Advanced Transformation Rules: Worked Example

This example demonstrates how to create an advanced transformation rule. Transformation rules are used at runtime to determine the internal representation of a product on an order line based on the information in the source order. An advanced rule can be used to compare two or more lines in an order.

This example shows you how to create a rule in which two lines are compared. If the first fulfillment line requires that you add an item and the second fulfillment line requires that you delete the same item, then the two actions cancel out one another.

1. Create the If statement: If the change in fulfillment line 1 is Add.
2. Create the If statement: If the change in fulfillment line 2 is Delete.
3. Create the If statement: If the inventory item in fulfillment line 1 is the same as the inventory item in fulfillment line 2.
4. Create the If statement: If the fulfillment line ID of fulfillment line 1 is different from the fulfillment ID of fulfillment line 2.
5. Create the Then statement: Then retract fulfillment line 1 and fulfillment line 2.

Creating an Advanced Transformation Rule

1. On the Manage Product Transformation Rules page, select Product Transformation Rules under Rulesets.
2. Click the New Rule icon.
3. Enter a rule name, such as "Consolidate Add and Delete Actions" in the Rule Name field.
4. In the Description field, type a description of the rule, such as "Rule to remove requests that cancel each other out."
5. Click the Expand icon.
6. Select the Advanced Mode check box.
7. In the If region, type a name, such as FulfillLine, in the Variable field.
8. In the Fact Type list of values, select FulfillLineVO.
9. On the next line, click the Search icon.
10. In the Condition Browser, click the Expand icon next to FulfillLine.
11. Select DeltaType.
12. Click **OK**.
13. In the Operator list of values, select Is.
14. In the Condition Browser, enter "Add."
15. Click **OK**.
16. Click the Add Pattern icon.
17. In the Variable field, enter a name, such as FulfillLine2.
18. In the Fact Type list, select FulfillLineVO.
19. Click Insert Test.
20. Click the Search icon.
21. In the Condition Browser, click the Expand icon next to FulfillLine2.
22. Select DeltaType.
23. Click **OK**.

24. In the Operator list of values, select Is.
25. Click the Search icon.
26. In the Condition Browser, click the Expand icon next to FulfillLine2.
27. In the Condition Browser field, enter "Delete."
28. Click **OK**.
29. Click the Insert Test icon.
30. Click the Search icon.
31. In the Condition Browser, click the Expand icon next to FulfillLine.
32. Select InventoryItemId.
33. Click **OK**.
34. In the Operator list of values, select Is.
35. Click the Search icon.
36. In the Condition Browser, click the Expand icon next to FulfillLine2.
37. Select InventoryItemId.
38. Click **OK**.
39. Click the Insert Test icon.
40. Click the Search icon.
41. In the Condition Browser, click the Expand icon next to FulfillLine.
42. Select FulfillLineId.
43. Click **OK**.
44. In the Operator list of values, select More Than.
45. Click the Search icon.
46. In the Condition Browser, click the Expand icon next to FulfillLine2.
47. Select FulfillLineId.
48. Click **OK**.
49. In the Then region, select Insert Action.
50. In the Action list of values, select Retract.
51. In the Target list of values, select FulfillLine.
52. Click the Insert Action icon.
53. In the Action list of values, select Retract.
54. In the Target list of values, select FulfillLine2.
55. Click **Save**.

Creating Posttransformation Defaulting Rules: Examples

Use posttransformation defaulting rules to automatically populate specific attributes onto an orchestration order based on the product transformation that

is applied to the orchestration order. Use these scenarios to understand how to use posttransformation defaulting rules.

Populating a Newly Added Orchestration Order Line with a Different Warehouse Attribute

Your company receives orders for laptop computers. Your product transformation rule transforms the sales order into an orchestration order with two lines:

- Orchestration order line 1: laptop computer
- Orchestration order line 2: alternating current adapter

You write a posttransformation defaulting rule that populates orchestration order line 2 with a warehouse that is different from the warehouse for the laptop computer.

Populating a New Orchestration Order with a New Attribute

Your company receives orders from that have the requested date as follows: MM/DD/YYYY. Your staff finds it useful to also know the day of the week because delivery options might be limited or cost more on certain days. You write a posttransformation defaulting rule that populates the day of the week onto the new orchestration order.

Define Transactional Business Intelligence Configuration

Define Transactional Business Intelligence Configuration: Highlights

Configure Oracle Transactional Business Intelligence for ad hoc reporting, review certain setup objects to be used in Transactional Business Intelligence, and manage the presentation catalog and currency type display.

Defining Transactional Business Intelligence Configuration

- Review details about the Transactional Business Intelligence tasks. Refer to the Oracle Fusion Transactional Business Intelligence Administrator's Guide.

Access to Person Data

Assigning Security Profiles to Job Roles for Oracle Fusion Transactional Business Intelligence Users: Explained

Users of Oracle Fusion Transactional Business Intelligence (Transactional Business Intelligence) need access to some person data for reporting purposes. To provide this access, you assign a predefined security profile to relevant job or abstract roles using the Oracle Fusion Human Capital Management (HCM) setup task Manage Data Role and Security Profiles. On completion of this task, Oracle Fusion Data Security is updated automatically for roles being used to access Transactional Business Intelligence.

Job or Abstract Roles and Related Security Profiles

The following table identifies, by Oracle Fusion product, the job and abstract roles that need access to person data and the predefined security profile that you assign to each role.

Product	Job or Abstract Role	Security Profile
Oracle Fusion Assets	Asset Accountant	View All Workers

Oracle Fusion Assets	Asset Accounting Manager	View All Workers
Oracle Fusion Enterprise Planning and Budgeting	Budget Analyst	View All Workers
Oracle Fusion Enterprise Planning and Budgeting	Budget Manager	View All Workers
Oracle Fusion Financial Consolidation Hub	Consolidation Accountant	View All Workers
Oracle Fusion Financial Consolidation Hub	Consolidation Manager	View All Workers
Oracle Fusion Financials Common Module	Intercompany Accountant	View All Workers
Oracle Fusion General Ledger	Financial Analyst	View All Workers
Oracle Fusion General Ledger	General Accountant	View All Workers
Oracle Fusion General Ledger	General Accounting Manager	View All Workers
Oracle Fusion Incentive Compensation	Incentive Compensation Participant Manager	View Manager Hierarchy
Oracle Fusion Inventory Management	Warehouse Manager	View All Workers
Oracle Fusion Project Foundation	Project Accountant	View All Workers
Oracle Fusion Project Foundation	Project Administrator	View All Workers
Oracle Fusion Project Foundation	Project Billing Specialist	View All Workers
Oracle Fusion Purchasing	Buyer	View All Workers
Oracle Fusion Sourcing	Category Manager	View All Workers
Oracle Fusion Sourcing	Sourcing Project Collaborator	View All Workers

For example, as part of their Transactional Business Intelligence setup:

- Oracle Fusion Assets implementors must assign the predefined security profile View All Workers to the Asset Accountant and Asset Accounting Manager job roles.
- Oracle Fusion Incentive Compensation implementors must assign the predefined security profile View Manager Hierarchy to the abstract role Incentive Compensation Participant Manager.

The security profiles that HCM roles need to access Transactional Business Intelligence are assigned during the setup of HCM data security: no additional setup is required for Transactional Business Intelligence purposes.

Enabling an Oracle Fusion Transactional Business Intelligence User to Access Person Data: Worked Example

This example shows how to assign a security profile to a job or abstract role to enable users with that role to access person data. This task is required for users of Oracle Fusion Transactional Business Intelligence (Transactional Business Intelligence) who do not also use Oracle Fusion Human Capital Management (HCM).

The following table summarizes key decisions for this scenario. When performing this task, use the job or abstract role for your product and the name of the relevant predefined person security profile in place of those shown here.

Decisions to Consider	In This Example
What is the name of the Transactional Business Intelligence job or abstract role?	Warehouse Manager
What is the name of the person security profile?	View All Workers

To perform these tasks, you must have the role IT Security Manager.

1. Launch the task Manage Data Role and Security Profiles.
2. Search for the job or abstract role.
3. Assign the relevant predefined security profile to the job or abstract role.

Launching the Task Manage Data Role and Security Profiles

1. On the Overview page of the Setup and Maintenance work area, click the **All Tasks** tab.
2. In the Search region, complete the fields as shown in this table:

Field	Value
Search	Tasks
Name	Manage Data Role and Security Profiles

3. Click **Search**.
4. In the search results, click **Go to Task** for the Manage Data Role and Security Profiles task.

Searching for the Job or Abstract Role

1. On the Manage HCM Data Roles page, enter the job name Warehouse Manager in the **Role** field.
2. Click **Search**.
3. In the search results, highlight the entry for the Warehouse Manager job role.
4. Click **Assign**.

Assigning the Security Profile to the Job Role

1. In the **Person Security Profile** field on the Assign Data Role: Security Criteria page, select the security profile View All Workers.
2. Click **Review**.
3. On the Assign Data Role: Review page, click **Submit**.

Define Extensions for Order Orchestration

Define Custom Enterprise Scheduler Jobs for Order Orchestration

Managing Job Definitions: Highlights

Oracle Enterprise Scheduler jobs are run in Oracle Fusion Applications to process data and, in some cases, to provide report output. A job definition contains the metadata that determines what the job does and what options are available to users who run the job. You can create and maintain job definitions for use in Oracle Fusion Applications.

Managing job definitions is fully described in the Oracle Fusion Applications Administrator's Guide. As you read content from that guide, note that the guide describes managing Oracle Enterprise Scheduler, including job definitions, from Oracle Enterprise Manager Fusion Applications Control. You can also access job definitions by starting in the Setup and Maintenance Overview page and searching for the Enterprise Scheduler job tasks for your applications.

Selecting the Appropriate Implementation Task

Each Enterprise Scheduler job definition task uses one Java EE application, which is referenced in the task name. You must use the right task because, to access the product job definition that you want to view or work on, the view objects must be contained in the application. If you do not select the right task, then the job definition will not be displayed properly or function correctly. The application name is usually the same as the product that the job definition belongs to, but not always.

- For example, the Oracle Fusion Payables Java EE application contains the Oracle Fusion Expenses product view objects. To create or maintain a job definition for use in Expenses, you select the Manage Custom Enterprise Scheduler Jobs for Payables and Related Applications task.
- In another example, the Oracle Fusion Payments product view objects are contained in both Oracle Fusion Payables and Oracle Fusion Receivables Java EE applications. You need to select the task appropriate to the job definition for Payments. Use the Manage Custom Enterprise Scheduler

Jobs for Receivables and Related Applications task if the job is for receivables functionality, or the Manage Custom Enterprise Scheduler Jobs for Payables and Related Applications task if the job is for payables functionality.

- Use the task description to see the products that correspond to the Java EE application specified in the task name. For example, the description for the Payables task lists Oracle Fusion Payables, Assets, Expenses, and Payments.
- You can view task descriptions in the help window for the task, if any, or in the generated setup task lists and tasks report from the Getting Started page.
- If you have access to the Manage Task Lists and Tasks page, you can also open the details for specific tasks to see the description.
- For general information about product and Java EE application relationships, use Oracle Enterprise Manager Fusion Applications Control (Fusion Applications Control).

See: Topology Section

Viewing and Creating Job Definitions

- You can access predefined and custom job definitions.

See: Viewing Job Definitions

- You can create jobs based on Java, PL/SQL, or any other supported technology.

See: Creating a Job Definition

- If you are using the Setup and Maintenance work area, then the Enable submission from Enterprise Manager check box is available for the job definition.
 - If you do not select this check box, then the job cannot be run from Enterprise Manager.
 - If you select this check box, then you can define parameters for this job definition only in Enterprise Manager. Save the rest of your work on the job definition, and then go to Enterprise Manager if you need to define parameters.

Managing List of Values Sources: Highlights

A list of values source for Oracle Enterprise Scheduler job definitions determines where a list of values comes from and what the specific values are. These lists of values are used in parameters and user properties of job definitions. For example, you can use a source of country values for a Country job parameter.

Managing list of values sources is fully described in the Oracle Fusion Applications Administrator's Guide. As you read content from that guide, note that the guide describes managing Oracle Enterprise Scheduler, including list of values sources, from Oracle Enterprise Manager Fusion Applications

Control. You can also access list of values sources by starting in the Setup and Maintenance Overview page and searching for Enterprise Scheduler job tasks.

Registering and Searching for List of Values Sources

- Create list of values sources to register them for use in job definitions.

See: Registering Sources for Lists of Values

- Search for list of values sources to edit or delete, or to make sure a particular source does not already exist before you create it.

See: Searching for List of Value Sources

Customizing Pages Using Page Composer: Highlights

You can customize dashboards and some work areas, where available, for all or some users based on a selected customization layer, for example only for users in a specific country or with a specific job role. When you select to customize a page from the **Administration** menu in the global area, you invoke Page Composer, which enables the customization. The Administration menu is only available if you have appropriate roles.

From the same menu, you can also:

- Customize the global area.
- Manage customizations by accessing a list of components in the current page and details about the layers in which each component is customized.
- Access sandboxes to make customizations to a runtime use session before deploying your changes to the mainline.

Customizing pages using Page Composer, managing customizations, and using sandboxes are described in the Oracle Fusion Applications Extensibility Guide. For more information on customization layers, see assets with the Customization Layer type in Oracle Enterprise Repository for Oracle Fusion Applications (<http://fusionappsoer.oracle.com>).

Editing Pages

- Customizations that you can make to existing pages include adding content and changing layout.

See: Editing a Page in Page Composer

- You can also update display and other options for specific components within the page.

See: Editing Component Properties in Page Composer

Global Page Template

- Customize the global area at the site layer.

See: Editing the UI Shell Template Used by All Pages

Managing Customizations

- Use the Manage Customizations dialog box to analyze and diagnose customization metadata, and to perform customization related tasks that cannot be done in the user interface, for example to revert customizations to a previous version. You can also do direct customization by manipulating the metadata and uploading it back.

See: Viewing and Diagnosing Runtime Customizations

Sandboxes

- Create or select an appropriate sandbox, and set it as active to capture your customizations using Page Composer. When you are ready, you publish the sandbox to make your changes available to users.

See: Using the Sandbox Manager

Importing and Exporting Setup Data

Configuration Packages: Explained

Almost all Oracle Fusion application implementations require moving functional setup data from one instance into another at various points in the lifecycle of the applications. For example, one of the typical cases in any enterprise application implementation is to first implement in a development or test application instance and then deploy to a production application instance after thorough testing. You can move functional setup configurations of applications from one application instance into another by exporting and importing Configuration packages from the Manage Configuration Packages page.

A Configuration Package contains the setup import and export definition. The setup import and export definition is the list of setup tasks and their associated business objects that identifies the setup data for export as well as the data itself. When you create a configuration package only the setup export and import definition exists. Once you export the configuration package appropriate setup data is added to the configuration package using the definition. Once a configuration package is exported, the setup export and import definition is locked and cannot be changed.

You generate the setup export and import definition by selecting an implementation project and creating a configuration package. The tasks and their associated business objects in the selected implementation project define the setup export and import definition for the configuration package. In addition, the sequence of the tasks in the implementation project determine the export and import sequence.

Exporting and Importing Setup Data: Explained

A configuration package is required to export setup data. You can export a configuration package once you create it, or at any time in the future. During export, appropriate setup data will be identified based on the setup export definition and added to the configuration package. The setup data in the configuration package is a snapshot of the data in the source application instance at the time of export. After the export completes, you can download the configuration package as a zipped archive of multiple XML files, move it to the target application instance, and upload and import it.

Export

You can export a configuration package multiple times by creating multiple versions. While the export definition remains the same in each version, the setup

data can be different if you modified the data in the time period between the different runs of the export process. Since each version of the configuration package has a snapshot of the data in the source instance, you can compare and analyze various versions of the configuration package to see how the setup data changed.

Import

In the target application instance, the setup import process will insert all new data from the source configuration package that does not already exist and update any existing data with changes from the source. Setup data that exists in the target instance but not in source will remain unchanged.

Export and Import Reports

You can review the results of the export and import processes using reports. The results appear ordered by business objects and include information on any errors encountered during the export or import process. If a setup export or import process paused due to errors encountered or for a manual task to be performed outside of the application, then you can resume the paused process.

These reports show what setup data was exported or imported and by which specific process. You can change the reports to validate the setup data as well as to compare or analyze it. A report is generated for each business object. These reports show the same information as the export and import results seen directly in the application.

Process status details are available as text files showing the status of an export or import process including the errors encountered during the process.

Moving Common Reference Objects

Moving Common Reference Objects: Overview

The common reference objects in Oracle Fusion Middleware Extensions for Applications are used by several setup tasks in the Setup and Maintenance work area. The common reference objects become a part of the configuration package that is created for an implementation project. While moving the application content, for example, from the test phase to the production phase of an implementation, you must pay special attention to the nuances of these common reference objects.

Parameters

The common reference objects are represented as business objects. A single object can be referenced in multiple setup tasks with different parameters. In the configuration package that is created for the implementation project, parameters passed to a setup task are also passed to the business objects being moved. As a result, the scope of the setup tasks is maintained intact during the movement.

Dependencies

Common reference objects may have internal references or dependencies among other common reference objects. Therefore, it is necessary that all the dependencies are noted before the movement of objects so that there are no broken references among the objects.

Business Objects for Moving Common Reference Objects: Points to Consider

Common reference objects in Oracle Fusion Functional Setup Manager are represented by business objects. These business objects are the agents that contain the application content and carry them across whenever the application setup is moved from one environment to another, for example, test environment to production environment.

Choice of Parameters

The following table lists the business objects, the corresponding movement details, and the effect of the setup task parameter on the scope of the movement.

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.

Business Object Name	Moved Functional Item	Effect on the Scope of Movement
Application Message	Messages and associated tokens	No parameters: all messages are moved. moduleType/moduleKey: only messages belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved. messageName/applicationId: only the specified message is moved.
Application Taxonomy	Application taxonomy modules and components	No parameters: all taxonomy modules and components are moved.
Application Attachment Entity	Attachment entities	No parameters: all attachment entities are moved. moduleType/moduleKey: only attachment entities belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.
Application Attachment Category	Attachment categories and category-to-entity mappings	No parameters: all attachment categories and category-to-entity mappings are moved. moduleType/moduleKey: only attachment categories belonging to the specified module and its descendant modules in the taxonomy hierarchy along with the respective category-to-entity mappings are moved.

Application Document Sequence Category	Document sequence categories	<p>No parameters: all categories are moved.</p> <p><code>moduleType/moduleKey</code>: only categories belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>code/applicationId</code>: only the specified document sequence category code is moved.</p>
Application Document Sequence	Document sequences and their assignments	<p>No parameters: all sequences are moved.</p> <p><code>moduleType/moduleKey</code>: only document sequences belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved</p> <p><code>name</code>: only the specified document sequence is moved.</p>
Application Descriptive Flexfield	Descriptive flexfield registration data and setup data	<p>No parameters: all descriptive flexfields are moved.</p> <p><code>moduleType/moduleKey</code>: only descriptive flexfields belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>descriptiveFlexfieldCode/applicationId</code>: only the specified descriptive flexfield is moved.</p> <hr/> <p>Note</p> <p>Importing a flexfield's metadata can change its deployment status and therefore, the affected flexfields must be redeployed. The import process automatically submits affected flexfields for redeployment.</p> <hr/> <p>Note</p> <p>Only flexfields with a deployment status of Deployed or Deployed to Sandbox are eligible to be moved.</p>

Application Extensible Flexfield	Extensible flexfield registration data and setup data, including categories	<p>No parameters: all extensible flexfields are moved</p> <p><code>moduleType/moduleKey</code>: only extensible flexfields belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>extensibleFlexfieldCode/applicationId</code>: only the specified extensible flexfield is moved.</p> <hr/> <p>Note</p> <p>Importing a flexfield's metadata can change its deployment status and therefore, the affected flexfields must be redeployed. The import process automatically submits affected flexfields for redeployment.</p> <hr/> <p>Note</p> <p>Only flexfields with a deployment status of Deployed or Deployed to Sandbox are eligible to be moved.</p>
Application Key Flexfield	Key flexfield registration data and setup data	<p>No parameters: all key flexfields are moved.</p> <p><code>moduleType/moduleKey</code>: only key flexfields belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>keyFlexfieldCode/applicationId</code>: only the specified key flexfield is moved.</p> <hr/> <p>Note</p> <p>Importing a flexfield's metadata can change its deployment status and therefore, the affected flexfields must be redeployed. The import process automatically submits affected flexfields for redeployment.</p> <hr/> <p>Note</p> <p>Only flexfields with a deployment status of Deployed or Deployed to Sandbox are eligible to be moved.</p>

Application Flexfield Value Set	Value set setup data	<p>No parameters: all value sets are moved.</p> <p><code>moduleType/moduleKey</code>: only value sets belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>valueSetCode</code>: only the specified value set is moved.</p> <hr/> <p>Note</p> <p>Importing a value set's metadata can change the deployment status of flexfields that use the value set, and therefore the affected flexfields must be redeployed. The import process automatically submits affected flexfields for redeployment.</p>
Application Reference Currency	Currency data	No parameters: all currencies are moved.
Application Reference ISO Language	ISO language data	No parameters: all ISO languages are moved.
Application Reference Industry	Industry data including industries in territories data	No parameters: all industries are moved.
Application Reference Language	Language data	No parameters: all languages are moved.
Application Reference Natural Language	Natural language data	No parameters: all natural languages are moved.
Application Reference Territory	Territory data	No parameters: all territories are moved.
Application Reference Time zone	Time zone data	No parameters: all time zones are moved.
Application Standard Lookup	Standard lookup types and their lookup codes	<p>No parameters: all standard lookups are moved.</p> <p><code>moduleType/moduleKey</code>: only standard lookups belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>lookupType</code>: only the specified common lookup is moved.</p>

Application Common Lookup	Common lookup types and their lookup codes	<p>No parameters: all common lookups are moved.</p> <p><code>moduleType/moduleKey</code> - only common lookups belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>lookupType</code>: only the specified common lookup is moved.</p>
Application Set-Enabled Lookup	Set-enabled lookup types and their lookup codes	<p>No parameters: all set-enabled lookups are moved.</p> <p><code>moduleType/moduleKey</code>: only set-enabled lookups belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>lookupType</code>: only the specified set-enabled lookup is moved.</p>
Application Profile Category	Profile categories	<p>No parameters: all profile categories are moved.</p> <p><code>moduleType/moduleKey</code>: only categories belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>name/applicationId</code>: only the specified category is moved.</p>
Application Profile Option	Profile options and their values	<p>No parameters: all profile options and their values are moved.</p> <p><code>moduleType/moduleKey</code>: only profile options and their values belonging to the specified module are moved.</p> <p><code>profileOptionName</code>: only the specified profile option and its values are moved.</p>

Application Profile Value	Profile options and their values	<p>No parameters: all profiles and their values are moved.</p> <p><code>moduleType/moduleKey</code>: only profiles and their values belonging to the specified module are moved.</p> <p><code>categoryName/ categoryApplicationId</code>: only profiles and their values belonging to the specified category are moved.</p> <p><code>profileOptionName</code>: only the specified profile and its values are moved.</p>
Application Reference Data Set	Reference data sets	No parameters: all sets are moved.
Application Reference Data Set Assignment	Reference data set assignments	<p><code>determinantType</code>: only assignments for the specified determinant type are moved.</p> <p><code>determinantType/ referenceGroupName</code>: only assignments for the specified determinant type and reference group are moved.</p>
Application Tree Structure	Tree structures and any labels assigned to the tree structure	<p>No parameters: all tree structures (and their labels) are moved.</p> <p><code>moduleType/moduleKey</code>: only tree structures (and their labels) belonging to the specified module are moved.</p> <p><code>treeStructureCode</code>: only the specified tree structure (with its labels) is moved.</p>
Application Tree	Tree codes and versions	<p>No parameters: all trees are moved.</p> <p><code>moduleType/moduleKey</code>: only trees belonging to the specified module are moved.</p> <p><code>treeStructureCode</code>: only trees belonging to the specified tree structure are moved.</p> <p><code>TreeStructureCode/TreeCode</code>: only trees belonging to the specified tree structure and tree code are moved.</p>

Application Tree Label	Tree structures and any labels assigned to the tree structure	<p>No parameters: all tree structures (and their labels) are moved.</p> <p><code>moduleType/moduleKey</code>: only tree structures (and their labels) belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>treeStructureCode</code>: only the specified tree structure (with its labels) is moved.</p>
Application Data Security Policy	Database resources, actions, conditions, and data security policies	<p>No parameters: all database resources/actions/conditions/policies are moved.</p> <p><code>moduleType/moduleKey</code>: only database resources/actions/conditions/policies belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</p> <p><code>objName</code>: only the specified database resource along with its actions/conditions/policies is moved.</p> <hr/> <p>Note</p> <ul style="list-style-type: none"> • 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.

External Integration

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.

Web Service Type	Description
ADF services	<p>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.</p> <p>Examples of ADF services include:</p> <ul style="list-style-type: none"> • <code>Worker.changeHireDate</code> - a service that updates the hire date of the worker business object. • <code>ProjectTask.createTask</code> - a service that adds a task to the project task business object.
Composite services	<p>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:</p> <ul style="list-style-type: none"> • <code>ProjectStatusChangeApproval.process</code> - a service that accepts the change in project status. • <code>ScheduleOrchestrationOrderFulfillmentLineService.scheduleOrder</code> - a service that schedules resources used to fulfill an order.

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: Highlights

The File Import and Export page accesses repositories of content. For example, each Oracle Fusion Applications instance connects to a single Oracle WebCenter Content server for content management.

The following documents describe use and administration of content management:

- Oracle WebCenter Content User's Guide for Content Server
- Oracle WebCenter Content System Administrator's Guide for Content Server

Using Content Management

- For information about what objects to upload and download, including templates for external data integration, refer to the Oracle Enterprise Repository for Oracle Fusion Applications.
- For general access to content management, including to all metadata and to manage accounts, use the Oracle WebCenter Content Server's standard service user interface.

See: Oracle WebCenter Content User's Guide for Content Server

- For information on creating accounts in WebCenter Content accounts, refer to WebCenter Content System Administrator's Guide for Content Server.

See: Accounts

- For information about naming accounts involved with import and export, see Files for Import and Export: Points to Consider.
- For programmatic upload and download to content management, refer to Oracle WebCenter Content System Administrator's Guide for Content Server.

See: About Batch Loading

- For information about the Oracle WebCenter Content Server Document Transfer Utility, see documentation resources in Oracle Enterprise Repository for Oracle Fusion Applications (<http://fusionappsoer.oracle.com>).

Security in Content Management

- For information about security, see the Security tab in Oracle Enterprise Repository for Oracle Fusion Applications.
- For information about roles such as the integration specialist roles for each product family, see the Oracle Fusion Applications security reference manuals for each offering. For example:

See: Oracle Fusion Applications Common Security Reference Manual

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
- File size

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.

Search criteria on the page are limited to the minimum metadata of content management records needed for file import and export.

Navigation: **Navigator > Tools > 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 Management 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 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 management 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 Oracle WebCenter 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 Introdoc 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

In the File Import and Export page, you can delete one file at a time. To delete multiple files at a time from content management, use the content management server's standard service user interface.

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.

Accessing Templates to Prepare External Data: 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 the templates
 - Opening XLS templates
 - Opening XML templates

Finding Templates and Control Files

To find the templates and control files:

1. Sign in to Oracle Enterprise Repository.
2. Use the fields in the Search pane to find the templates.
3. Enter the information in the following table to search by scheduled process :

Field	Value
Search String	FBDI
Type	FusionApps: Scheduled Process

Note

You can further narrow the search by logical business area.

4. Click Search.
The search displays a Load Interface File for Import job.

Downloading the Templates

To download the templates:

1. Select the Load Interface File for Import job in the upper pane and then click the Documentation tab in the lower pane.
Notice the list of links to application-specific import jobs.
2. Click a link to access the job.
3. Click the Documentation tab.
4. Click the links at the bottom of the tab to 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

- XML templates for Oracle Data Integrator

Opening the XLS Template

If you intend to prepare your data in a spreadsheet format, download the XLS template:

1. Open the XLS template. The first worksheet in each template 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.

Opening the XML Template

If you intend to prepare your data in using Oracle Data Integrator (ODI), download the XML template into ODI:

1. Import the family-level template as a model folder.
2. Import the product-level template as a model folder within the family-level model folder.
3. Import the product template as a model within the product-level model folder
4. Create the integration project.
5. Create the package.
6. Add and configure these elements:
 - a. The integration projects
 - b. ODIZip tool
 - c. The content management document transfer utility (Currently not available in Oracle Cloud implementations.)
7. Execute the package.

The package generates the CSV file and compresses it into a ZIP file.

Note

In Oracle Cloud implementations, you must upload the ZIP file to the content management server 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.

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. By following the instructions with exactness when you use the templates, you 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 expected 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 server 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 to 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

Using XML Integration Templates to Generate Data Files: Points to Consider

Use XML templates in Oracle Data Integrator to prepare your external data for loading and importing. Oracle Enterprise Repository for Oracle Fusion Applications includes three types of XML templates that you import as target models in your Oracle Data Integrator repository.

Oracle Enterprise Repository includes these three levels of XML files:

- Family-level
- Product-level
- Product

Family-Level XML Files

A family-level XML file is common to a group of product-level model folders and product models.

Consider the following points when you use family-level XML files:

- The family-level XML file supports all of the Oracle Enterprise Repository assets in the family, for example Oracle Fusion Financials or Human Capital Management.
- You import the family-level XML file into your Oracle Data Integrator repository prior to importing the other XML files.
- You import one family-level XML file as a model folder for each family of products.
- You import each family-level XML file as a top-level model folder.
- You import the family-level XML file one time; it supports all subsumed product-level model folders.
- You select Synonym mode Insert Update as the import type.

Product-Level XML Files

A product-level XML file is common to a group of product models.

Consider the following points when you use product-level XML files:

- The product-level XML file supports all of the Oracle Enterprise Repository assets in the product line, for example Fixed Assets, General Ledger, or Payables.
- You import one product-level XML file as a model folder for each line of products.
- You import the product-level XML file as a model folder into your Oracle Data Integrator repository after you import the family-level XML file, but before you import product XML files.
- You import each product-level XML file as a midlevel model folder within the appropriate family-level model folder.
- You import the product-level XML file one time; it supports all subsumed product models.
- You select Synonym mode Insert Update as the import type.

Product XML Files

A product XML file represents a specific Oracle Enterprise Repository interface table asset.

Consider the following points when you use product XML files:

- You import one product XML file as a model for each interface table or set of tables, for example Mass Additions.
- You import the product XML file as a model into your Oracle Data Integrator repository after you import the product-level XML file.
- You import each product XML file as a model within the appropriate product-level model folder.
- You import each product XML file one time.
- You select Synonym mode Insert Update as the import type.
- The model is based on File technology.
- After you import the product model, you connect the model to the correct logical schema.

Transferring Data Files to Content Management: Explained

After you generate the comma-separated values (CSV) file, transfer it to the content management server.

You can use the following methods to transfer files:

- File Import and Export page in Oracle Fusion Applications

- Oracle WebCenter Content Document Transfer Utility
- Oracle web service

For details about programmatic file transfer using the Oracle WebCenter Content Document Transfer Utility, refer to documentation resources in Oracle Enterprise Repository for Oracle Fusion Applications (<http://fusionappsoer.oracle.com>).

Aspects of transferring data files to content management involve the following:

- Target accounts
- Accessing transferred content

Target Accounts

You must transfer files to the predefined account in content management that corresponds to the interface table or assets.

Interface Table	Predefined Account
Payables Standard Invoice Import	fin/payables/import
<ul style="list-style-type: none"> • AutoInvoice Import • Receivables Standard Receipt Import • Customer Import • China Value Added Tax Invoice Import 	fin/receivables/import
<ul style="list-style-type: none"> • BAI2 Format Bank Statements Import • EDIFACT FINSTA Format Bank Statements Import • ISO200022 CAMT053 Format Bank Statements Import • SWIFT MT940 Format Bank Statements Import 	fin/cashManagement/import
<ul style="list-style-type: none"> • Fixed Asset Mass Additions Import • Fixed Asset Mass Adjustments Import • Fixed Asset Mass Retirements Import • Fixed Asset Mass Transfers Import • Fixed Asset Units of Production Import 	fin/assets/import
Intercompany Transaction Import	fin/intercompany/import
<ul style="list-style-type: none"> • Journal Import • Chart Of Account Segment Values and Hierarchies Import 	fin/generalLedger/import
General Ledger Budget Balance Import	fin/budgetBalance/import
Supplier Bank Account Import	fin/payables/import
Tax Configuration Content Import	fin/tax/import
Import Blanket Purchase Agreements	prc/blanketPurchaseAgreement/import
Import Contract Purchase Agreements	prc/contractPurchaseAgreement/import

Import Purchase Orders	prc/purchaseOrder/import
Import Requisitions	prc/requisition/import
<ul style="list-style-type: none"> • Import Suppliers • Import Supplier Sites • Import Supplier Site Contacts • Import Supplier Site Assignments 	prc/supplier/import
Project Enterprise Resource Import	prj/projectManagement/import
Project Unprocessed Expenditure Item Import	prj/projectCosting/import
Cycle Count Import	scm/cycleCount/import
Inventory Reservation Import	scm/inventoryReservation/import
Inventory Transaction Import	scm/inventoryTransaction/import
Item Import	scm/item/import
Receiving Receipt Import	scm/receivingReceipt/import
Shipment Request Import	scm/shipmentRequest/import

You can create subaccounts for further file organization. However you need to create the subaccount under 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 Management:

- 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.

Load Interface File for Import Process

Loads external setup or transaction data from the data file on the content management server to interface 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 management server.

Parameters

Import Process

Select the target import process.

Data file

Enter the relative path and the file name of the ZIP data file on the content management server.

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 relative path of the data file.

Note

The path is relative to the account in the content management server. If the file exists in the account, enter the file name only. If the file exists in a subaccount subordinate to the account, you must include all subaccounts and the file name.

- c. The data file is retained on the content management server when the process is complete.
6. Submit the process.

If no errors exist in the data file, then the process populates the interface tables.

Finding and Submitting the Import Process

Run the import process appropriate to your data to import the data into application data tables into the application 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.

Glossary

abstract role

A description of a person's function in the enterprise that is unrelated to the person's job (position), such as employee, contingent worker, or line manager. A type of enterprise role.

accounting flexfield

The chart of accounts that determines the structure, such as the number and order of individual segments, as well as the corresponding values per segment.

action

The kind of access named in a security policy, such as view or edit.

activity

User task that needs to be assigned and completed as part of the order fulfillment process.

ADF

Acronym for Application Developer Framework. A set of programming principles and rules for developing software applications.

application identity

Predefined application level user with elevated privileges. An application identity authorizes jobs and transactions for which other users are not authorized, such as a payroll run authorized to access a taxpayer ID while the user who initiated the job is not authorized to access such personally identifiable information.

application role

A role specific to applications and stored in the policy store.

Applications Core

Abbreviation for Oracle Fusion Middleware Extensions for Applications. The technical product code is FND.

assignment

A set of information, including job, position, pay, compensation, managers, working hours, and work location, that defines a worker's or nonworker's role in a legal employer.

assignment set

A group of sourcing rules, bills of distribution, or both, in which each rule or bill is assigned to an assignment level, and to attribute values for the attributes applicable to that assignment level. An assignment set defines a supply chain.

automatic assignment catalog

A non-hierarchical catalog to which categories that match the catalog's Catalog Structure value are automatically added. Add categories and share categories actions are disabled for this catalog configuration.

balancing segment

A chart of accounts segment used to automatically balance all journal entries for each value of this segment.

beneficiary

A person or organization designated to receive benefits from a compensation plan on the death of the plan participant.

bill of distribution

A multilevel specification of the means by which several, specific organizations can replenish items.

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.

browsing category

Parent or intermediate category that is associated with other categories in the catalog hierarchy, but has no assigned items.

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.

category

Catalog component that is associated to a catalog to classify items.

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.

contingent worker

A self-employed or agency-supplied worker. Contingent worker work relationships with legal employers are typically of a specified duration. Any person who has a contingent worker work relationship with a legal employer is a contingent worker.

continuous collection

An incremental, entity-specific method of collecting data from the Oracle Fusion source system into the order orchestration and planning data repository.

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.

cumulative manufacturing lead time

The total amount of time needed to manufacture an item, assuming there is no availability of any raw material or components. The manufacturing time is added up from the primary routing of the item and all the subassemblies beneath it.

cumulative total lead time

The total lead time of an assembly plus the largest adjusted cumulative total lead time of its components.

data dimension

A stripe of data accessed by a data role, such as the data controlled by a business unit.

data instance set

The set of human capital management (HCM) data, such as one or more persons, organizations, or payrolls, identified by an HCM security profile.

data role

A role for a defined set of data describing the job a user does within that defined set of data. A data role inherits job or abstract roles and grants entitlement to access data within a specific dimension of data based on data security policies. A type of enterprise role.

data role template

A template used to generate data roles by specifying which base roles to combine with which dimension values for a set of data security policies.

data security

The control of access to data. Data security controls what action a user can taken against which data.

data security policy

A grant of entitlement to a role on an object or attribute group for a given condition.

database resource

An applications data object at the instance, instance set, or global level, which is secured by data security policies.

department

A division of a business enterprise dealing with a particular area of activity.

descriptive flexfield

Customizable expansion space, such as fields used to capture additional descriptive information or attributes about an entity, such as customer cases. Information collection and storage may be configured to vary based on conditions or context.

determinant

A value that determines which reference data set will be used in a specific business context.

determinant type

Designates the field within transactional columns that controls how data is shared across organizations such as business unit, asset book, cost organization or project unit. The type determines the reference data sets that would be used in a transaction.

determinant type

An additional and optional field within transactional columns (besides category and application) that is used to assign document sequences. The available determinant types are Business Unit, Ledger, Legal Entity, and Tax Registration.

determinant value

A value specific to the determinant type dimension of a document sequence. The determinant value is relevant in a document sequence assignment only if the document sequence has a determinant type. If Ledger is the determinant type for a document sequence, the determinant value is the specific ledger number whose documents are numbered by the document sequence.

division

A business-oriented subdivision within an enterprise. Each division is organized to deliver products and services or address different markets.

document sequence

A unique number that is automatically or manually assigned to a created and saved document.

document type

A categorization of person documents that provides a set of options to control what document information to retain, who can access the documents, whether the documents require approval, and whether the documents are subject to expiry. A document type exists for a combination of document category and subcategory.

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 allow grouping contexts into categories.

feature choice

A selection you make when configuring offerings that modifies a setup task list, or a setup page, or both.

flexfield

Grouping of extensible data fields called segments, where each segment is an attribute added to an entity for capturing additional information.

flexfield segment

An extensible data field that represents an attribute on an entity and captures a single atomic value corresponding to a predefined, single extension column in the Oracle Fusion Applications database. A segment appears globally or based on a context of other captured information.

function security

The control of access to a page or a specific widget or functionality within a page. Function security controls what a user can do.

gallery

A searchable collection of portraits that combines the functions of the person directory with corporate social networking and self-service applications for both workers and managers.

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.

global entity

An entity for which a single value is stored for all source systems when instances of the entity are collected and stored in the order orchestration and planning data repository.

global sourcing rule

A sourcing rule that is applicable to every organization.

grade

A component of the employment model that defines the level of compensation for a worker.

HCM

Abbreviation for Human Capital Management.

HCM data role

A job role, such as benefits administrator, associated with specified instances of Oracle Fusion Human Capital Management (HCM) data, such as one or more positions or all persons in a department.

HCM securing object

An HCM object that secures access to both its own data and data in other, related objects. For example, access to a specified set of person records can allow access to data secured by person records, such as goal plans and evaluations.

identity

A person representing a worker, supplier, or customer.

interface table

A database table used for transferring data between applications or from an external application or data file.

inventory organization

An organization that tracks inventory transactions and balances, and can manufacture or distribute products.

item master

A collection of data that describes items and their attributes recorded in a database file.

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.

JAR

Abbreviation for JAVA archive.

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.

local sourcing rule

A sourcing rule that is applicable to one, and only one, specific organization.

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.

managed person

In Oracle Fusion Human Capital Management security, a person for whom the user can maintain some information. For example, line managers can maintain information about their direct and indirect reports, and workers can maintain information about themselves, their dependents, and their beneficiaries.

native catalog

A catalog that a user is managing.

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.

orchestration order

Fulfillment-oriented representation of a source sales order. When a sales order enters Oracle Fusion Distributed Order Orchestration it is transformed to an orchestration order.

orchestration process

A sequence of steps that takes one or more fulfillment lines through the fulfillment process.

order orchestration and planning data repository

The set of data collected from source systems and stored for use by order orchestration and order promising processes.

OWLCS

Abbreviation for Oracle WebLogic Communication Services. Offers the TPCC service to Fusion applications and sets up the calls via SIP integration with the telephony network.

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.

pending worker

A person who will be hired or start a contingent worker placement and for whom you create a person record that is effective before the hire or start date.

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.

person type

A subcategory of a system person type, which the enterprise can define. Person type is specified for a person at the employment-terms or assignment level.

personally identifiable information

Any piece of information that can potentially be used to uniquely identify, contact, or locate a single person. Within the context of an enterprise, some PII data can be considered public, such as a person's name and work phone number, while other PII data is confidential, such as national identifier or passport number.

PL/SQL

Abbreviation for procedural structured queried language.

portrait

A selection of information about a worker or nonworker, including contact details, social connections, and activities and interests, that can be viewed and edited. Both the amount and type of information and the available actions depend on the role of the portrait user.

position

A specific occurrence of one job, fixed within one department, also often one location. For example, the position Finance Manager is an instance of the job Manager in the Finance Department.

primary ledger

Main record-keeping ledger.

privilege

A grant or entitlement of access to functions and data. A privilege is a single, real world action on a single business object.

processing constraints

Rule that controls which changes can be made to an orchestration order, by whom, and when.

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.

public person

In Oracle Fusion Human Capital Management security, a person for whom some basic information is publicly available. For example, users typically access the contact details of public persons, such as phone numbers and locations, using the person gallery.

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.

referenced category

A category within the native catalog that is shared from a designated source catalog. A reference category is not editable.

resource

People designated as able to be assigned to work objects, for example, service agents, sales managers, or partner contacts. A sales manager and partner contact can be assigned to work on a lead or opportunity. A service agent can be assigned to a service request.

role

Controls access to application functions and data.

role hierarchy

Structure of roles to reflect an organization's lines of authority and responsibility. In a role hierarchy, a parent role inherits all the entitlement of one or more child roles.

role mapping

A relationship between one or more job roles, abstract roles, and data roles and one or more conditions. Depending on role-mapping options, the role can be provisioned to or by users with at least one assignment that matches the conditions in the role mapping.

role provisioning

The automatic or manual allocation of an abstract role, a job role, or a data role to a user.

sandbox

A runtime session that commits changes out of reach of mainline users.

security profile

A set of criteria that identifies one or more human capital management (HCM) objects of a single type for the purposes of securing access to those objects. Security profiles can be defined for persons, organizations, positions, countries, LDGs, document types, payrolls, payroll flows, and workforce business processes.

security reference implementation

Predefined function and data security in Oracle Fusion Applications, including role based access control, and policies that protect functions, data, and segregation of duties. The reference implementation supports identity management, access provisioning, and security enforcement across the tools, data transformations, access methods, and the information life cycle of an enterprise.

segregation of duties

An internal control to prevent a single individual from performing two or more phases of a business transaction or operation that could result in fraud.

set

Reference data that is organized into groups appropriate to organizational entities, to enable reference data sharing.

set enabled

An entity, such as a lookup, customer, location, organization, or document attachment, that is allowed to participate in reference data sharing by drawing on the data of a reference data set.

shared category

A category within a source catalog that has been added to a native catalog as a referenced category. The category can be shared with one or more catalogs.

SOA

Abbreviation for service-oriented architecture.

source system

Any point of origin with integrated, cohesive business application data from which fulfillment data, or planning data, or both, can be extracted into data files.

source-specific entity

An entity for which a record is stored for each instance in each source system when data is collected and stored in the order orchestration and planning data repository. The stored data includes which source system the record was collected from.

sourcing assignment level

The scope, or level of granularity, for which a sourcing rule or bill of distribution is applicable when assigned within a specific assignment set.

sourcing rule

A specification of the means by which organizations can replenish items.

SQL predicate

A type of condition using SQL to constrain the data secured by a data security policy.

staging tables upload

The final step in the extract, store in staging tables, and upload from staging tables, method of collecting data from an external source system into the order orchestration and planning data repository. The upload can be performed as a targeted or net change upload.

step

Building block of an orchestration process definition that specifies which task layer service to call or which subprocess to launch. Each step contains definitions of the task type, task, and service. Branching is also specified in the step. Tasks are made up of one or more steps.

system person type

A fixed name that the application uses to identify a group of people.

targeted collection

A periodic, full-refresh, entity-specific method of collecting data from an Oracle Fusion source system into the order orchestration and planning data repository. A targeted collection completely replaces all data in the data repository for each entity included in the collection cycle.

task

Representation of a fulfillment task that Oracle Fusion Distributed Order Orchestration requests in a fulfillment system, for example, Invoice or Reservation. A task can have multiple orchestration process steps.

territory

A legally distinct region that is used in the country field of an address.

total lead time

The time required to procure or manufacture an item. For manufactured assemblies, total lead time equals the time required to manufacture the item from its components on the primary routing, assuming infinite availability of components.

tree

Information or data organized into a hierarchy with one or more root nodes connected to branches of nodes. A tree must have a structure where each node corresponds to data from one or more data sources.

tree structure

Characteristics applied to trees, such as what data to include or how the tree is versioned and accessed.

tree version

An instance of a tree. If a tree is associated with a reference data set, all versions belong to one set. Includes life cycle elements such as start and end date and a status indicator whether the tree is active or not.

value set

A set of valid values against which values entered by an end user are validated. The set may be tree structured (hierarchical).

Web service upload

An incremental, entity-specific method of collecting data from an external source system into the order orchestration and planning data repository.

work relationship

An association between a person and a legal employer, where the worker type determines whether the relationship is a nonworker, contingent worker, or employee work relationship.

worker type

A classification selected on a person's work relationship, which can be employee, contingent worker, pending worker, or nonworker.

workforce business process

A defined sequence of tasks, registered for the enterprise, that delivers a major business goal. The tasks may be allocated to multiple roles across multiple products.

XML

Abbreviation for eXtensible markup language.

XML filter

A type of condition using XML to constrain the data secured by a data security policy.