# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Preface</strong></td>
<td>i</td>
</tr>
<tr>
<td><strong>1 Overview</strong></td>
<td>1</td>
</tr>
<tr>
<td>Overview of Common Implementation</td>
<td>1</td>
</tr>
<tr>
<td>Purchase and Activation of Oracle Cloud Application Services</td>
<td>1</td>
</tr>
<tr>
<td><strong>2 Maintain Common Reference Objects</strong></td>
<td>3</td>
</tr>
<tr>
<td>Overview of Common Reference Objects</td>
<td>3</td>
</tr>
<tr>
<td>Set General Preferences for All Users</td>
<td>3</td>
</tr>
<tr>
<td>Why can't I edit setup data?</td>
<td>3</td>
</tr>
<tr>
<td>How can I determine which languages appear in the Translation Editor?</td>
<td>4</td>
</tr>
<tr>
<td>How do I define whether the user image, name, or initials display in the global header?</td>
<td>4</td>
</tr>
<tr>
<td>Define Application Taxonomy</td>
<td>4</td>
</tr>
<tr>
<td>Define ISO Reference Data</td>
<td>6</td>
</tr>
<tr>
<td>Manage Audit Policies</td>
<td>8</td>
</tr>
<tr>
<td>Manage Oracle Social Network Objects</td>
<td>25</td>
</tr>
<tr>
<td>Define Global Search</td>
<td>29</td>
</tr>
<tr>
<td><strong>3 Define Currencies and Currency Rates</strong></td>
<td>39</td>
</tr>
<tr>
<td>Manage Currencies</td>
<td>39</td>
</tr>
<tr>
<td>Manage Conversion Rate Types</td>
<td>39</td>
</tr>
</tbody>
</table>
4 Define Enterprise Structures

Overview
Enterprise Structures Business Process Model
Guidelines for Configuring Global Enterprises
Model Your Enterprise Management Structure
Essbase Character and Word Limitations
Define Initial Configuration with the Enterprise Structures Configurator
Define Reference Data Sharing
Define Enterprise: Manage Enterprise HCM Information
Define Enterprise: Manage Locations
Define Geographies
Define Legal Jurisdictions and Authorities
Define Legal Entities: Manage Legal Entity
Define Legal Entities: Manage Legal Entity HCM Information
Define Legal Entities: Manage Legal Entity Tax Profile
Define Legal Entities: Define Legal Reporting Units
Define Business Units: Manage Business Units
Define Business Units: Assign Business Unit Business Function
Define Business Units: Manage Service Provider Relationships
Define Business Units: Specify Contract Management Business Function Properties

5 Define Security
Overview of Security

6 Define Approval Management
Overview of Approval Management
Access for Workflow Administrators
Define the Due Date and Expiration Policies for Workflow Tasks
Define People to Support Workflow Tasks
Assignments and Routing
Workflow Notifications
Manage Workflow Rules Using a Spreadsheet
Change Workflow Task Titles
Workflow Task Life Cycle Setup
Set Up the Worklist Region on My Dashboard
Best Practices for Approval Management
Workflow Transaction Console
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Define Help Configuration</td>
<td>193</td>
</tr>
<tr>
<td></td>
<td>Set Up Help</td>
<td>193</td>
</tr>
<tr>
<td></td>
<td>Show Help Icons by Default</td>
<td>193</td>
</tr>
<tr>
<td></td>
<td>Give People Access to Create and Edit Help</td>
<td>194</td>
</tr>
<tr>
<td></td>
<td>Enable Global Search for Added Help</td>
<td>194</td>
</tr>
<tr>
<td></td>
<td>Create Groups to Limit Access to Added Help</td>
<td>194</td>
</tr>
<tr>
<td>8</td>
<td>Define Application Toolkit Configuration</td>
<td>197</td>
</tr>
<tr>
<td></td>
<td>Overview of Application Toolkit Configuration</td>
<td>197</td>
</tr>
<tr>
<td></td>
<td>Map Reports to Work Areas</td>
<td>197</td>
</tr>
<tr>
<td></td>
<td>Set Watchlist Options</td>
<td>199</td>
</tr>
<tr>
<td>9</td>
<td>Define Enterprise Scheduler Job Definitions and Job Sets</td>
<td>203</td>
</tr>
<tr>
<td></td>
<td>Overview of Managing Job Definitions and Job Sets</td>
<td>203</td>
</tr>
<tr>
<td></td>
<td>Manage List of Values Sources</td>
<td>203</td>
</tr>
<tr>
<td></td>
<td>Managing Job Definitions</td>
<td>204</td>
</tr>
<tr>
<td></td>
<td>Job Sets</td>
<td>210</td>
</tr>
<tr>
<td>10</td>
<td>Define Applications Core Configuration</td>
<td>215</td>
</tr>
<tr>
<td></td>
<td>Overview of Applications Core Configuration</td>
<td>215</td>
</tr>
<tr>
<td></td>
<td>Import Flexfields, Lookups, or Profile Values Using Web Services</td>
<td>215</td>
</tr>
<tr>
<td></td>
<td>Define Lookups</td>
<td>220</td>
</tr>
<tr>
<td></td>
<td>Manage Messages</td>
<td>232</td>
</tr>
<tr>
<td></td>
<td>Define Document Sequences</td>
<td>237</td>
</tr>
<tr>
<td></td>
<td>Define Trees</td>
<td>240</td>
</tr>
<tr>
<td></td>
<td>Define Profile Options</td>
<td>254</td>
</tr>
<tr>
<td></td>
<td>Define Flexfields</td>
<td>259</td>
</tr>
<tr>
<td></td>
<td>Define Attachments</td>
<td>364</td>
</tr>
<tr>
<td>11</td>
<td>External Integration</td>
<td>369</td>
</tr>
<tr>
<td></td>
<td>Web Services</td>
<td>369</td>
</tr>
<tr>
<td></td>
<td>Files for Import and Export</td>
<td>376</td>
</tr>
<tr>
<td></td>
<td>External Data Integration Services for Oracle Cloud</td>
<td>378</td>
</tr>
<tr>
<td></td>
<td>Deep Links</td>
<td>389</td>
</tr>
</tbody>
</table>
12 Moving Common Reference Objects

Overview of Moving Common Reference Objects 391
Business Objects for Moving Common Reference Objects 391
Guidelines for Moving Related Common Reference Objects 397
Guidelines for Moving Common Reference Objects Using the Seed Data Framework 398
Preface

This preface introduces information sources that can help you use the application.

Using Oracle Applications

Help

Use help icons to access help in the application. If you don't see any help icons on your page, click your user image or name in the global header and select Show Help Icons. Not all pages have help icons. You can also access the Oracle Help Center to find guides and videos.

Watch: This video tutorial shows you how to find and use help.

You can also read about it instead.

Additional Resources

- **Community**: Use Oracle Cloud Customer Connect to get information from experts at Oracle, the partner community, and other users.
- **Training**: Take courses on Oracle Cloud from Oracle University.

Conventions

The following table explains the text conventions used in this guide.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates user interface elements, navigation paths, or values you enter or select.</td>
</tr>
<tr>
<td><strong>monospace</strong></td>
<td>Monospace type indicates file, folder, and directory names, code examples, commands, and URLs.</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than symbol separates elements in a navigation path.</td>
</tr>
</tbody>
</table>
Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website. Videos included in this guide are provided as a media alternative for text-based help topics also available in this guide.

Contacting Oracle

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit My Oracle Support or visit Accessible Oracle Support if you are hearing impaired.

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! You can send an e-mail to: oracle_fusion_applications_help_ww_grp@oracle.com.
Overview of Common Implementation

Common implementation involves performing setup tasks that are common and available within multiple offerings. The Application Extensions and other functional areas comprise these common setup and implementation tasks.

Application Extensions

Use the Application Extensions functional area to configure common business objects. For example, this functional area comprises tasks that help you to:

- Set help options, for example to make help icons visible to everyone.
- Review and manage objects, for example currencies and reference data sets that are shared across applications.
- Configure common reference objects such as flexfields, document sequences, and profile options that affect the functionality and look of Oracle Applications Cloud.

Other Functional Areas

Other functional areas contain several tasks to manage common functions that apply to the entire implementation. Examples of such functional areas include Legal Structures, Users and Security, and Enterprise Profile. Use these functional areas to, for example:

- Set up security, enterprise structures, geographies, and business units.
- Create and maintain user accounts and synchronize the list of users and roles stored in Lightweight Directory Access Protocol (LDAP).

Purchase and Activation of Oracle Cloud Application Services

Purchase and activate Oracle Cloud Application Services from the Oracle Cloud web site, https://cloud.oracle.com. The process involves:

- Purchasing and activating your services.
- Verifying that the services are activated, monitoring the services, and performing other administrative tasks.

Purchasing and activating any Oracle Cloud service is described in the Oracle Cloud: Getting Started with Oracle Cloud guide. Many administrative tasks are covered in the Oracle Cloud: Managing and Monitoring Oracle Cloud guide. All of the following references in this section point to one of these guides.
**Note:** Not everything in these guides is relevant to Oracle Cloud Application Services, for example details about Oracle Java Cloud Service and Oracle Database Cloud Service.

### Terminology and User Roles
- Before you proceed, you should understand terms that are used in documentation about Oracle Cloud. Refer to the Oracle Cloud: Getting Started with Oracle Cloud guide.
  
  See: Oracle Cloud Terminology

- You should also be familiar with roles for users of the Oracle Cloud web site or of the actual service. Refer to the Oracle Cloud: Getting Started with Oracle Cloud guide.

  See: Oracle Cloud User Roles and Privileges

### Service Purchase and Activation
- Your buyer or an Oracle sales representative orders a cloud service and specifies information about the account administrator during the ordering process. Refer to the Oracle Cloud: Getting Started with Oracle Cloud guide.

  See: Buy an Oracle Cloud Subscription

- The account administrator receives an e-mail with a link to activate the service. As part of activation, the account administrator provides information about the administrator who performs the functions of both the service administrator and the identity domain administrator. Refer to the Oracle Cloud: Getting Started with Oracle Cloud guide.

  See: Activate Your Order from Your Welcome Email

### Service Verification
- The administrator who was identified during the activation process:
  - Verifies that the service is activated. Refer to the Oracle Cloud: Getting Started with Oracle Cloud guide.

    See: Verifying That Your Services Are Ready

  - Manages and monitors the service. Refer to the Oracle Cloud: Managing and Monitoring Oracle Cloud guide.

    See: My Account Administration

    See: Performing Service-Specific Tasks

  - Optionally create initial administrator and implementation users before enterprise structures setup.

    • Alternatively, the administrator sets up enterprise structures and then creates service users, including functional implementors.

    • Functional implementors perform configuration and setup steps.

    • Developers can add features to extend the application.
2 Maintain Common Reference Objects

Overview of Common Reference Objects

The Maintain Common Reference Objects task list contains tasks that support implementation of common functionality, such as data security, reference data sets, or general preferences.

Use this task list to manage common reference objects that are defined centrally and shared across applications. You can search for and access this task list in the Setup and Maintenance work area.

To make the Maintain Common Reference Objects task list available in your implementation project, go to the Offerings work area and enable the Maintain Common Reference Objects feature.

Related Topics
- Overview of Moving Common Reference Objects

Set General Preferences for All Users

Use the Global Admin Preferences page to set general preferences, such as language, currency, and time zone for all users. Use this task only if you want to update preferences for all users. Your users can still set preferences for themselves using the General Preferences page. Once you set the general preferences for all users, if specific users don't change their preferences, then the preferences that you have set for all users will apply to them.

To set general preferences:

1. In the Setup and Maintenance work area, go to the Set User General Preferences task.
2. Set the required general preferences for your users.
3. If you only want to set the preferences for new users whose preferences haven't been set by them yet, select the Reset preferences for new users only check box. Selecting this option excludes all users whose preferences were set at some point in time.
4. Click Save.

Related Topics
- How can I set general preferences for myself

Why can't I edit setup data?

The configuration of your setup data may be protected. Application developers mark some configurations as protected, indicating that you can't edit them.

Some examples of configurations that may be protected are:

- Descriptive flexfields
- Extensible flexfield contexts
- Extensible flexfield pages
How can I determine which languages appear in the Translation Editor?

Use the Translation Editor Languages profile option to specify the languages as the profile values, so that they are available as options for the user on the Translated Editor dialog box.

1. In the Setup and Maintenance work area, go to the Manage Administrator Profile Values task.
2. Search for the Translation Editor Languages profile option.
3. In the Profile Values section, specify the required languages as its profile values, separated by comma. You can either list the complete names of the languages or just the language codes specified on the Manage Languages page, for example, F for French, KO for Korean.

   **Note:** You can specify only those languages that are available in the supported language pack. If you don't specify any value, all available language packs are supported.

4. Click **Save and Close**. The language entries will take effect for the user in the next sign in.

**Related Topics**

- Enter or Edit Translated Text

How do I define whether the user image, name, or initials display in the global header?

Set the User Image Display Enabled (FND_USER_PHOTO_ENABLED) profile option. If you select:

- Yes, then the user image is displayed in the **global header** only if the user uploaded a photo. Otherwise, the user's initials are displayed. By default, this profile option is set to Yes.
- No, then only the user name is displayed in the global header.

**Related Topics**

- Change Your Photo

Define Application Taxonomy
Overview of Application Taxonomy

Oracle application components and functions are organized in a hierarchy, ranging from product lines to logical business areas. The hierarchy represents a breakdown of products into units based on how applications are installed and supported.

In the Application Extensions functional area within your offering, search for the Manage Taxonomy Hierarchy task and view the hierarchy on the Manage Taxonomy Hierarchy page.

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 life cycle.

  See: How to Manage the Life cycle

Modules in Application Taxonomy

The highest level of the hierarchy is product line, followed by the 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. A module is a node at any of these levels. Each level is briefly described here.

- Product Line: A collection of products associated with a single brand name, for example, Oracle Fusion.
- Product Family: A collection of products associated with a functional area that may or may not be licensed together as a single unit, for example Oracle Fusion Financials.
- Application: A single product within a product family, containing closely related features for a specific business solution, for example General Ledger.
- Logical Business Area: 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 up to two or three levels deep.

Guidelines for Managing Modules in Application Taxonomy

In the application taxonomy hierarchy, when you create a module, it becomes a child of the currently selected node. Once created, you can't delete the module or move it elsewhere in the hierarchy.

From the Manage Taxonomy Hierarchy page, navigate to the Create Child Module or Edit Module page to manage the modules. 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, a unique read-only ID is automatically generated. The module contains two other identifiers: Module key and alternative 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 additional identifiers are provided for the product line, product family, and application modules. However, you can optionally add them for logical business areas and new modules.

Note: Don’t 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 Oracle Application.

Names
Module name is the logical name for the module. The name must be unique among nodes within the hierarchy level with the same parent, but Oracle recommends keeping it unique in the entire hierarchy. The user name and description can appear to users in other parts of Oracle Applications Cloud.

Usage Types
Though you can update the usage type to reflect the current state of the module, just doing so doesn’t affect the actual state. For example, setting a module as installed doesn’t mean the module is actually installed if the installation itself didn’t take place. Installation refers to operations related to laying down all the components required to create an Oracle Applications Cloud environment. 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 data residing in 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 the 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 Applications Cloud Functional Core (ASK) tables.

Define ISO Reference Data

Considerations for Natural Languages
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 can’t be deleted, but 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 must 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?
The predefined territories are countries from the International Organization for Standardization (ISO) 3166 standard. Edit territory descriptions to determine how they are displayed in lists of country values in an application. You don’t have to edit territory names or codes unless there is a specific requirement. Create territories if new countries emerge and the application isn’t yet patched with the latest ISO country values.

Note: The National Language Support (NLS) territory codes are territory identifiers used in the application. Don’t edit the codes unless you must change the association between ISO and the application territory.

When do I create or edit industries?
To meet a specific business need, you may edit industry names or descriptions of industries except for those belonging to the North American Industry Classification System (NAICS). Edit the industry descriptions also to determine how they appear in an application.

You may also create industries with specifications not included in the NAICS standards.

When do I associate industries with territories?
To meet specific business needs, you can associate industries with territories. For example, administrators can configure a page in different ways for different sets of users of the same industry, but residing in different countries.

When do I create or enable currencies?
Create or enable any currency for displaying monetary amounts, assigning currency to ledgers, entering transactions, recording balances, or for any reporting purpose. All currencies listed in the International Organization for Standardization (ISO) 4217 standard are supported.

The default currency is set to United States Dollar (USD).

Related Topics
• Considerations for Defining Currencies
What's the difference between precision, extended precision, and minimum accountable unit for a currency?

Precision refers to the number of digits placed after the decimal point used in regular currency transactions. For example, USD would have 2 as the precision value for transactional amounts, such as $1.00.

Extended precision is the number of digits placed after the decimal point and must be greater than or equal to the precision value. For calculations requiring greater precision, you can enter an extended precision value such as 3 or 4. That would result in the currency appearing as $1.279 or $1.2793.

Minimum accountable unit is the smallest denomination for the currency. For example, for USD that would be .01 for a cent.

You can set these values for a currency using the **Manage Currencies** task in the Application Extensions functional area in the Setup and Maintenance work area.

What's a statistical unit currency type?

The statistical unit currency type denotes the Statistical (STAT) currency used to record financial statistics in the financial reports, allocation formulas, and other calculations.

When do I create or edit ISO languages?

Edit the names and descriptions of International Organization for Standardization (ISO) languages to determine how they appear in the application. The ISO languages are a part of the ISO 639 standard. If any change to the ISO standard doesn't reflect in the application, you can update the ISO alpha-2 code or add languages to provide up-to-date information.

When do I edit languages?

Installed languages automatically appear on the Manage Languages page. This page also displays all languages that are available for installation and translation. Each dialect is treated as a separate language.

Generally, you don't need to edit any of the detailed fields unless absolutely necessary.

Can I add or edit time zones?

You usually don't add or edit time zones because all standard time zones are provided. However, you may create time zones if new zones become standard and the application isn't yet updated with the latest values. You can rename existing time zones and enable them. Only the enabled time zones are available for all users to select while setting their regional general preferences.

To add or edit time zones, use the following in the Setup Maintenance work area:

- Functional Area: Application Extensions
- Task: Manage Time Zones

Manage Audit Policies

Audit Policies

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 required for reporting.

Enabling Audit Functionality

For Oracle Applications Cloud, 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 and manage audit, ensure that you have a role with the assigned privilege Manage Audit Policies (FND_MANAGE_AUDIT_POLICIES_PRIV). For appropriate assignment of roles and privileges, check with your security administrator.

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, see Audit Events for Oracle Applications Cloud Middleware (Doc ID 2114143.1) on My Oracle Support at https://support.oracle.com.

If you don't want an application to be audited, you can stop the audit process by setting the Audit Level option to None.

Related Topics

- Audit Events for Oracle Applications Cloud Middleware

Audit Configuration for Business Object Attributes

Audit enables tracking 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 operations performed on an object and its attributes, such as create, update, and delete. To configure audit business object attributes, use the Manage Audit Policies task in the Setup and Maintenance work area.

Selecting an Application

To set up auditing, you must select a web application that contains the required business objects that can be audited. From the list of business objects, select those business objects 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. If the object selected in an audit hierarchy is also a part of several other audit hierarchies, the attribute configuration for that object is applicable to all the hierarchies in that application.

Tip: For business objects based on flexfields, select the Flexfields (Additional Attributes) check box to view and add or remove flexfield attributes, to include or exclude them from the audit.

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 Applications Cloud 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 aren't audited. To continue to audit the business object with select attributes, deselect those attributes that aren't to be audited. When users view the audit history for an application, they can specify the period for which they want the results. Therefore, make a note of when you start and stop auditing an application.

For example, users intend to view the audit history of an object for the previous week, but auditing for that object was stopped last month. They wouldn't get any audit results for that week, because during the entire month that object wasn't audited. Even if you enable audit for that object today, users can't get the wanted results because audit data until today isn't available.

Overview of Audit Configuration

To set up auditing for Oracle Applications Cloud, select the Manage Audit Policies task from the Setup and Maintenance work area within your offering. To set up auditing for Oracle Fusion Middleware products, select the level of auditing mapped to a predefined set of metadata and the events that have to be audited. Information about 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 as audit-specific assets that you can use to create the config.xml configuration file. To get a list of audit-specific assets, see Audit Events for Oracle Applications Cloud Middleware (Doc ID 2114143.1) on My Oracle Support at https://support.oracle.com.

Oracle Fusion Middleware Products


  See: Auditing Web Services

Oracle Fusion Security Products

- Configure business objects to enable auditing in Oracle Fusion security products. Refer to Oracle Fusion Middleware Application Security Guide.

  See: Oracle Fusion Middleware Audit Framework Reference

Related Topics

- Audit Events for Oracle Applications Cloud Middleware

Using Auditing to Monitor Application Changes

You use auditing to monitor user activity and all configuration, security, and data changes that have been made to an application. You can enable business objects to allow auditing, recording, and retrieving information about when the objects were created, modified, and removed.
The following table shows the business objects you can enable for auditing.

<table>
<thead>
<tr>
<th>Area</th>
<th>Parent Objects</th>
<th>Child Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Collections</td>
<td>Aging Buckets</td>
<td>• Aging Method Buckets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Aging Method Bucket Details</td>
</tr>
<tr>
<td>Advanced Collections</td>
<td>Backup Collectors</td>
<td>• Backup Collectors</td>
</tr>
<tr>
<td>Advanced Collections</td>
<td>Collection Preferences</td>
<td>• Collection Preferences</td>
</tr>
<tr>
<td>Advanced Collections</td>
<td>Collections Scoring and Strategy</td>
<td>• Collections Scoring and Strategy Assignments</td>
</tr>
<tr>
<td></td>
<td>Assignments</td>
<td></td>
</tr>
<tr>
<td>Advanced Collections</td>
<td>Collections Scoring Formula</td>
<td>• Collections Scoring Formula</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Collections Scoring Data Points</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Collections Scoring Data Point Details</td>
</tr>
<tr>
<td>Advanced Collections</td>
<td>Collections Strategies</td>
<td>• Collections Strategies Group</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Collections Strategies</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Collections Strategy Tasks</td>
</tr>
<tr>
<td>Advanced Collections</td>
<td>Collectors</td>
<td>• Collectors</td>
</tr>
<tr>
<td>Advanced Collections</td>
<td>Dunning Configurations</td>
<td>• Dunning Configurations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Dunning Configuration Details</td>
</tr>
<tr>
<td>Assets</td>
<td>Fixed Asset Bonus Rule</td>
<td>• Fixed Asset Bonus Rule</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Bonus Rate</td>
</tr>
<tr>
<td>Assets</td>
<td>Fixed Asset Book</td>
<td>• Fixed Asset Book</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tax Authority Name</td>
</tr>
<tr>
<td>Assets</td>
<td>Fixed Asset Calendar</td>
<td>• Fixed Asset Calendar</td>
</tr>
<tr>
<td>Assets</td>
<td>Fixed Asset Cash Generating Unit</td>
<td>• Fixed Asset Cash Generating Unit</td>
</tr>
<tr>
<td>Assets</td>
<td>Fixed Asset Category</td>
<td>• Fixed Asset Category</td>
</tr>
<tr>
<td>Assets</td>
<td>Fixed Asset Category Alias</td>
<td>• Fixed Asset Category Alias</td>
</tr>
<tr>
<td>Area</td>
<td>Parent Objects</td>
<td>Child Objects</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>Assets</td>
<td>Fixed Asset Ceiling</td>
<td>• Ceiling Type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Fixed Asset Ceiling</td>
</tr>
<tr>
<td>Assets</td>
<td>Fixed Asset Depreciation Method</td>
<td>• Fixed Asset Depreciation Method</td>
</tr>
<tr>
<td>Assets</td>
<td>Fixed Asset Distribution Set</td>
<td>• Fixed Asset Distribution Set</td>
</tr>
<tr>
<td>Assets</td>
<td>Fixed Asset Fiscal Year</td>
<td>• Fixed Asset Fiscal Year</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Fiscal Year</td>
</tr>
<tr>
<td>Assets</td>
<td>Fixed Asset Key</td>
<td>• Fixed Asset Key</td>
</tr>
<tr>
<td>Assets</td>
<td>Fixed Asset Key Alias</td>
<td>• Fixed Asset Key Alias</td>
</tr>
<tr>
<td>Assets</td>
<td>Fixed Asset Location</td>
<td>• Fixed Asset Location</td>
</tr>
<tr>
<td>Assets</td>
<td>Fixed Asset Location Alias</td>
<td>• Fixed Asset Location Alias</td>
</tr>
<tr>
<td>Assets</td>
<td>Fixed Asset Price Index</td>
<td>• Fixed Asset Price Index</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Price Index Value</td>
</tr>
<tr>
<td>Assets</td>
<td>Fixed Asset Prorate Convention</td>
<td>• Convention Type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Fixed Asset Prorate Convention</td>
</tr>
<tr>
<td>Assets</td>
<td>System Controls</td>
<td>• System Controls</td>
</tr>
<tr>
<td>Bill Management</td>
<td>Bill Management System Options</td>
<td>• Bill Management System Parameters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Bill Management System Option Preferences</td>
</tr>
<tr>
<td>Budgetary Control</td>
<td>Budgetary Control and Encumbrance</td>
<td>• Enabled Ledger</td>
</tr>
<tr>
<td></td>
<td>Accounting</td>
<td>• Enabled Business Unit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Enabled Business Function</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Enabled Transaction Source for Project</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Accounting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Enabled Transaction Type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Enabled Journal Source</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Enabled Journal Category</td>
</tr>
<tr>
<td>Budgetary Control</td>
<td>Control Budget</td>
<td>• Control Budget</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Control Budget Structure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Budget Segment Tree</td>
</tr>
<tr>
<td>Area</td>
<td>Parent Objects</td>
<td>Child Objects</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Budget Override Rule</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Budget Override Journal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Budget Override Transaction Type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Budget Override Rule User Assignment</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Supplemental Rule</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Supplemental Rule Filter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Supplemental Rule Filter Value Detail</td>
</tr>
<tr>
<td>Cash Management</td>
<td>Ad Hoc Payee Information</td>
<td>• Ad Hoc Payee Information</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• External Payees Audit Setup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ External Payees</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Bank Account Usage</td>
</tr>
<tr>
<td>Cash Management</td>
<td>Bank Account</td>
<td>• Bank Account Payment Document</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Bank Account Checkbook</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Bank Account Use</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Bank Account General Ledger Account</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Payment Document Category</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Bank Account Grant</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Bank Account Transaction Creation Rule</td>
</tr>
<tr>
<td>Cash Management</td>
<td>Bank Statement Parsing Rule Set</td>
<td>• Bank Statement Parsing Rule Set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Bank Statement Parsing Rule</td>
</tr>
<tr>
<td>Cash Management</td>
<td>Bank Statement Reconciliation Matching Rule Set</td>
<td>• Bank Statement Reconciliation Matching Rule Set Item</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Bank Statement Reconciliation Matching Rule</td>
</tr>
<tr>
<td>Cash Management</td>
<td>Bank Statement Transaction Creation Rule</td>
<td>• Bank Statement Transaction Creation Rule</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Bank Transaction Codes</td>
</tr>
<tr>
<td>Cash Management</td>
<td>Bank Transaction Codes</td>
<td>• Bank Transaction Codes</td>
</tr>
<tr>
<td>Area</td>
<td>Parent Objects</td>
<td>Child Objects</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Cash Management</td>
<td>Cash Positioning and Forecasting Options</td>
<td>• Cash Positioning and Forecasting Options</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Cash Positioning and Forecasting Time Periods</td>
</tr>
<tr>
<td>Cash Management</td>
<td>Cash Positioning and Forecasting Transaction Grouping</td>
<td>• Cash Position Cube Dimensions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Cash Position Dimension Sources</td>
</tr>
<tr>
<td>Cash Management</td>
<td>Cash Transaction Type Mapping</td>
<td>• Cash Transaction Type Mapping</td>
</tr>
<tr>
<td>Data Security</td>
<td>Data Security Access for Users</td>
<td>• User Role Data Assignments</td>
</tr>
<tr>
<td>Expenses</td>
<td>Manage Conversion Rates and Policies</td>
<td>• Edit Conversion Rates and Policies</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Individual Currency Exceptions</td>
</tr>
<tr>
<td>Expenses</td>
<td>Manage Expense Report Templates</td>
<td>• Default Template</td>
</tr>
<tr>
<td>Expenses</td>
<td>Manage Expenses System Options</td>
<td>• System Options</td>
</tr>
<tr>
<td>Financials Common Module</td>
<td>Additional Intercompany Balancing and Clearing Options</td>
<td>• Additional Intercompany Balancing and Clearing Options Setup</td>
</tr>
<tr>
<td>Financials Common Module</td>
<td>Business Unit</td>
<td>• Business Functions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Service Provider Relationships</td>
</tr>
<tr>
<td>Financials Common Module</td>
<td>Intercompany Balancing Rules</td>
<td>• Intercompany Rule Setup</td>
</tr>
<tr>
<td>Financials for Americas</td>
<td>Collection Remittance Batch Template</td>
<td>• Bank Transfer Select Control</td>
</tr>
<tr>
<td>Financials for Americas</td>
<td>Local Holiday for Brazil</td>
<td>• Local Holiday Sets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Local Holiday Dates</td>
</tr>
<tr>
<td>Financials for Americas</td>
<td>Receivables Bank Instruction for Brazil</td>
<td>• Receivables Bank Instruction for Brazil</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Account Combinations</td>
<td>• Account Combinations Setup</td>
</tr>
<tr>
<td>Area</td>
<td>Parent Objects</td>
<td>Child Objects</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Accounting and Reporting Sequences</td>
<td>• Sequences Header</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Sequences Version</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Sequences Context</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Sequences Assignment</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Accounting Automation Setup</td>
<td>• Accounting Automation Notifications</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Accounting Automation Notification Users</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Accounting Calendars</td>
<td>• Accounting Calendar Setup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Period Details</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Period Status</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Allocations</td>
<td>• Allocation Rule Sets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Allocation Runtime Prompt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Allocation Rules</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Allocation Runtime Prompt</td>
</tr>
<tr>
<td>General Ledger</td>
<td>AutoPost Criteria Sets</td>
<td>• AutoPost Criteria Set Setup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ AutoPost Criteria Set Options</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Chart of Accounts Mapping</td>
<td>• Chart of Accounts Mapping Rule</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Segment Rule</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Rollup Rule</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Account Rule</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Clearing Accounts Reconciliation Rules</td>
<td>• Clearing Accounts Reconciliation Rules Setup</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Currency Conversion Rate Type and Rates</td>
<td>• Conversion Rate Type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Conversion Rate Rule</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Daily Rates</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Data Access Set</td>
<td>• Data Access Set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Data Access Set Assignment</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Encumbrance Carry Forward Rules</td>
<td>• Encumbrance Carry Forward Rules Setup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Encumbrance Carry Forward Accounts</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Encumbrance Types</td>
<td>• Journal Encumbrance Type</td>
</tr>
<tr>
<td>Area</td>
<td>Parent Objects</td>
<td>Child Objects</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Journal Categories</td>
<td>• Journal Category Setup</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Journal Reversal Criteria Sets</td>
<td>• Journal Reversal Criteria Set Setup</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Journal Sources</td>
<td>• Journal Source Setup</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Journals</td>
<td>• Journal Batch</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ Journal Header</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Journal Line</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Ledger Set</td>
<td>• Ledger Set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ Ledger Set Normal Assignment</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Ledgers</td>
<td>• Ledger Configuration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ Legal Entities</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Legal Entity Balancing Segment Values</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Revaluations</td>
<td>• Revaluations Setup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ Revaluation Ranges</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Statistical Units of Measure</td>
<td>• Statistical Units of Measure Setup</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Suspense Accounts</td>
<td>• Suspense Account Setup</td>
</tr>
<tr>
<td>Intercompany</td>
<td>Intercompany Customer Supplier Association</td>
<td>• Customer and Supplier Association Setup</td>
</tr>
<tr>
<td>Intercompany</td>
<td>Intercompany Organization</td>
<td>• Organization Setup</td>
</tr>
<tr>
<td>Area</td>
<td>Parent Objects</td>
<td>Child Objects</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>Intercompany</td>
<td>Intercompany Receivables Assignment</td>
<td>• Receivables Assignment Setup</td>
</tr>
<tr>
<td>Intercompany</td>
<td>Intercompany System Option</td>
<td>• System Option Setup</td>
</tr>
<tr>
<td>Intercompany</td>
<td>Intercompany Transaction Type</td>
<td>• Transaction Type Setup</td>
</tr>
<tr>
<td>Legal Entity Configurator</td>
<td>Legal Association</td>
<td>• Associated Business Entities Setup</td>
</tr>
<tr>
<td>Legal Entity Configurator</td>
<td>Legal Contact</td>
<td>• Legal Contact Role Setup</td>
</tr>
<tr>
<td>Legal Entity Configurator</td>
<td>Legal Entity</td>
<td>• Legal Entity Setup</td>
</tr>
<tr>
<td>Legal Entity Configurator</td>
<td>Legal Entity and Legal Reporting Unit Registration</td>
<td>• Legal Registration Setup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Legal Registration Function Setup</td>
</tr>
<tr>
<td>Legal Entity Configurator</td>
<td>Legal Jurisdiction</td>
<td>• Legal Jurisdiction Setup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Legal Jurisdiction Function Setup</td>
</tr>
<tr>
<td>Legal Entity Configurator</td>
<td>Legal Reporting Unit</td>
<td>• Legal Reporting Unit Setup</td>
</tr>
<tr>
<td>Payables</td>
<td>Aging Periods</td>
<td>• Aging Periods</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Aging Period Details</td>
</tr>
<tr>
<td>Payables</td>
<td>Bank Charges</td>
<td>• Bank Charge Details</td>
</tr>
<tr>
<td>Payables</td>
<td>Common Options for Payables and Procurement</td>
<td>• Common Options for Payables and Procurement</td>
</tr>
<tr>
<td>Payables</td>
<td>Distribution Sets</td>
<td>• Distribution Sets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Distribution Sets Details</td>
</tr>
<tr>
<td>Payables</td>
<td>General Payables Options</td>
<td>• General Payables Options</td>
</tr>
<tr>
<td>Payables</td>
<td>Income Tax Regions</td>
<td>• Income Tax Regions</td>
</tr>
<tr>
<td>Payables</td>
<td>Interest Rates</td>
<td>• Interest Rates</td>
</tr>
<tr>
<td>Area</td>
<td>Parent Objects</td>
<td>Child Objects</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>Payables</td>
<td>Invoice</td>
<td>• Invoice</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Invoice Installments</td>
</tr>
<tr>
<td>Payables</td>
<td>Invoice Holds and Releases</td>
<td>• Invoice Holds and Releases</td>
</tr>
<tr>
<td>Payables</td>
<td>Invoice Tolerances</td>
<td>• Invoice Tolerances</td>
</tr>
<tr>
<td>Payables</td>
<td>Payables Calendars</td>
<td>• Payables Calendars  • Period Information</td>
</tr>
<tr>
<td>Payables</td>
<td>Payment Terms</td>
<td>• Payment Terms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Set Assignments</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Installments</td>
</tr>
<tr>
<td>Payables</td>
<td>Reporting Entities</td>
<td>• Reporting Entities</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Balancing Segment Values</td>
</tr>
<tr>
<td>Payments</td>
<td>Credit Card Brands Audit Setup</td>
<td>• Acceptable Credit Card Brands</td>
</tr>
<tr>
<td>Payments</td>
<td>Disbursement Payment Methods Audit Setup</td>
<td>• Disbursement Payment Methods  • Disbursement Payment Methods Usage</td>
</tr>
<tr>
<td>Payments</td>
<td>Disbursement System Options Audit Setup</td>
<td>• Disbursement System Options Enterprise Level  • Disbursement System Options Business Unit Wise</td>
</tr>
<tr>
<td>Payments</td>
<td>External Bank Account Audit Setup</td>
<td>• External Bank Account  • External Bank Account Owner  • Intermediary Bank Account</td>
</tr>
<tr>
<td>Payments</td>
<td>External Payees Audit Setup</td>
<td>• External Payees  • Bank Account Usage</td>
</tr>
<tr>
<td>Payments</td>
<td>External Payers Audit Setup</td>
<td>• External Payers  • Intermediary Bank Account</td>
</tr>
<tr>
<td>Payments</td>
<td>Formats Audit Setup</td>
<td>• Format</td>
</tr>
<tr>
<td>Payments</td>
<td>Funds Capture Payment Method Audit Setup</td>
<td>• Funds Capture Payment Method  Attribute Applicability</td>
</tr>
<tr>
<td>Area</td>
<td>Parent Objects</td>
<td>Child Objects</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Payments</td>
<td>Funds Capture Process Profile Audit Setup</td>
<td>• Funds Capture Payment Method</td>
</tr>
<tr>
<td>Payments</td>
<td>Internal Payees Audit Setup</td>
<td>• Routing Rule</td>
</tr>
<tr>
<td>Payments</td>
<td>Payment Codes Audit Setup</td>
<td>• Payment Code</td>
</tr>
<tr>
<td>Payments</td>
<td>Payment Process Profile Audit Setup</td>
<td>• Applicable Payment Process Profile</td>
</tr>
<tr>
<td>Payments</td>
<td>Payment Process Profile for Bank Account</td>
<td>• Payment Instruction Creation Rule</td>
</tr>
<tr>
<td>Payments</td>
<td>Payment Process Profile for Credit Card</td>
<td>• Payment Creation Rule</td>
</tr>
<tr>
<td>Payments</td>
<td>Payment Process Profile for Debit Card</td>
<td>• Payment Grouping Rule</td>
</tr>
<tr>
<td>Payments</td>
<td>Payment Process Profile for Debit Card</td>
<td>• Remit Advice</td>
</tr>
<tr>
<td>Payments</td>
<td>Payment System Account</td>
<td>• Payment System Account</td>
</tr>
<tr>
<td>Payments</td>
<td>Payment System Audit Setup</td>
<td>• Payment Process Profile</td>
</tr>
<tr>
<td>Payments</td>
<td>System Security Options Audit Setup</td>
<td>• System Security Options</td>
</tr>
<tr>
<td>Payments</td>
<td>Transmission Configurations Audit Setup</td>
<td>• Transmission Configuration</td>
</tr>
<tr>
<td>Payments</td>
<td>Validation Sets</td>
<td>• Transmission Value</td>
</tr>
<tr>
<td>Payments</td>
<td>Validation Sets</td>
<td>• Validation Values</td>
</tr>
<tr>
<td>Payments</td>
<td>Validation Assignments</td>
<td>• Validation Assignments</td>
</tr>
<tr>
<td>Receivables</td>
<td>Application Rule Sets</td>
<td>• Application Rule Sets</td>
</tr>
<tr>
<td>Receivables</td>
<td>Approval Limits</td>
<td>• Receivables User Approval Limit</td>
</tr>
<tr>
<td>Receivables</td>
<td>AutoAccounting Rules</td>
<td>• AutoAccounting Rules Segment</td>
</tr>
<tr>
<td>Area</td>
<td>Parent Objects</td>
<td>Child Objects</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Receivables</td>
<td>AutoCash Rule Sets</td>
<td>• AutoCash Rule Details</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o AutoCash Rules</td>
</tr>
<tr>
<td>Receivables</td>
<td>AutoInvoice Grouping Rules</td>
<td>• AutoInvoice Grouping Rule</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o AutoInvoice Grouping Rule Transaction Type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o AutoInvoice Grouping Rule Group By</td>
</tr>
<tr>
<td>Receivables</td>
<td>AutoInvoice Line Ordering Rules</td>
<td>• AutoInvoice Line Ordering Rule</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o AutoInvoice Line Ordering Rule Order By</td>
</tr>
<tr>
<td>Receivables</td>
<td>AutoMatch and Application Exception Rules</td>
<td>• Application Exception Rules</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Application Exception Rule Details</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o AutoMatch Rule Set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o AutoMatch Rule Set Details</td>
</tr>
<tr>
<td>Receivables</td>
<td>Balance Forward Billing Cycles</td>
<td>• Receivables Balance Forward Billing Cycle</td>
</tr>
<tr>
<td>Receivables</td>
<td>Credit Scoring Models</td>
<td>• Credit Scoring Model Details</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Credit Scoring Model Range</td>
</tr>
<tr>
<td>Receivables</td>
<td>Customer Profile</td>
<td>• Trading Community Customer Billing Account</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Trading Community Customer Account Relationship</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Trading Community Customer Account or Site Role</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Trading Community Customer Account or Site Role Responsibility</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Trading Community Customer Account Site</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Trading Community Customer Account Site Use</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Customer Profile</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Customer Profile Amount</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Trading Community Address Purpose</td>
</tr>
<tr>
<td>Area</td>
<td>Parent Objects</td>
<td>Child Objects</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Receivables</td>
<td>Interest Tiers</td>
<td>• Interest Tier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Interest Tier Levels Information</td>
</tr>
<tr>
<td></td>
<td>Late Charge Schedules</td>
<td>• Late Charge Schedule</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Late Charge Schedule Header</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Late Charge Schedule Line</td>
</tr>
<tr>
<td></td>
<td>Payment Terms</td>
<td>• Payment Terms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Payment Terms Line</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Payment Terms Line Discount</td>
</tr>
<tr>
<td></td>
<td>Receipt Classes and Methods</td>
<td>• Receipt Classes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Receipt Method</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Receipt Method Account</td>
</tr>
<tr>
<td></td>
<td>Receipt Sources</td>
<td>• Receipt Sources</td>
</tr>
<tr>
<td></td>
<td>Receivables Activities</td>
<td>• Receivables Activities</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Receivables Activities Legal Entity Detail</td>
</tr>
<tr>
<td></td>
<td>Receivables Customer Profile Classes</td>
<td>• Receivables Customer Profile Class</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Receivables Customer Profile Class Amount</td>
</tr>
<tr>
<td></td>
<td>Reference Accounts</td>
<td>• Transaction Type Reference Account</td>
</tr>
<tr>
<td></td>
<td>Remit-to Addresses</td>
<td>• Remit to Address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Receipts from Criteria</td>
</tr>
<tr>
<td></td>
<td>Revenue Contingencies</td>
<td>• Revenue Contingencies</td>
</tr>
<tr>
<td></td>
<td>Revenue Scheduling Rules</td>
<td>• Revenue Scheduling Rules</td>
</tr>
<tr>
<td></td>
<td>Standard Memo Lines</td>
<td>• Standard Memo Lines</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Transaction Type Reference Account</td>
</tr>
<tr>
<td></td>
<td>Standard Messages</td>
<td>• Standard Messages</td>
</tr>
<tr>
<td>Area</td>
<td>Parent Objects</td>
<td>Child Objects</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Receivables</td>
<td>Statement Cycles</td>
<td>• Statement Cycles</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Statement Cycle Dates</td>
</tr>
<tr>
<td>Receivables</td>
<td>System Options</td>
<td>• System Options</td>
</tr>
<tr>
<td>Receivables</td>
<td>Transaction Sources</td>
<td>• Transaction Sources</td>
</tr>
<tr>
<td>Receivables</td>
<td>Transaction Types</td>
<td>• Transaction Type Reference Account</td>
</tr>
<tr>
<td>Receivables</td>
<td>Transmission Formats for Lockbox</td>
<td>• Transmission Formats for Lockboxes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Transmission Record Formats for Lockboxes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Transmission Field Formats for Lockboxes</td>
</tr>
<tr>
<td>Regional Localizations</td>
<td>Fiscal Document Business Unit</td>
<td>• Fiscal System Options</td>
</tr>
<tr>
<td>Regional Localizations</td>
<td>Fiscal Document Default and Control</td>
<td>• Fiscal System Options</td>
</tr>
<tr>
<td>Regional Localizations</td>
<td>Fiscal Document Series</td>
<td>• Fiscal Document Series</td>
</tr>
<tr>
<td>Revenue Management</td>
<td>Contract Identification Rules</td>
<td>• Contract Identification Rules</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Contract Identification Rule Details</td>
</tr>
<tr>
<td>Revenue Management</td>
<td>Implied Performance Obligation</td>
<td>• Implied Performance Obligation Templates</td>
</tr>
<tr>
<td></td>
<td>Templates</td>
<td>◦ Implied Performance Obligation Template Details</td>
</tr>
<tr>
<td>Revenue Management</td>
<td>Item Groups</td>
<td>• Item Groups</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Item Group Assignments</td>
</tr>
<tr>
<td>Revenue Management</td>
<td>Performance Obligation Identification Rules</td>
<td>• Performance Obligation Identification Rules</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Performance Obligation Grouping Rules</td>
</tr>
<tr>
<td>Area</td>
<td>Parent Objects</td>
<td>Child Objects</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Revenue Management</td>
<td>Performance Obligation Templates</td>
<td>• Performance Obligation Templates</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Performance Obligation Grouping Rule Details</td>
</tr>
<tr>
<td></td>
<td>Pricing Dimension Assignments</td>
<td>• Pricing Dimension Assignment Names</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Pricing Dimension Assignment Names</td>
</tr>
<tr>
<td></td>
<td>Pricing Dimension Bands</td>
<td>• Pricing Dimension Band Names</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Pricing Dimension Bands</td>
</tr>
<tr>
<td></td>
<td>Source Document Types</td>
<td>• Source Document Types</td>
</tr>
<tr>
<td></td>
<td>Standalone Selling Price Effective Periods</td>
<td>• Standalone Selling Price Effective Periods</td>
</tr>
<tr>
<td></td>
<td>Standalone Selling Price Profiles</td>
<td>• Standalone Selling Price Profiles</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Standalone Selling Price Profile Assignments</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ Standalone Selling Price Profile Price Band Assignments</td>
</tr>
<tr>
<td></td>
<td>System Options for Revenue Management</td>
<td>• Source Document Type Integration with Revenue Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Currency Conversion for Revenue Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Receivables Transaction Source Integration with Revenue Management</td>
</tr>
<tr>
<td>Subledger Accounting</td>
<td>Account Rule Service</td>
<td>• Account Rule Details Audit</td>
</tr>
<tr>
<td>Subledger Accounting</td>
<td>Accounting Method Service</td>
<td>• Journal Entry Rule Set Assignment Audit</td>
</tr>
<tr>
<td>Subledger Accounting</td>
<td>Condition Service</td>
<td>• Conditions Audit</td>
</tr>
<tr>
<td>Subledger Accounting</td>
<td>Description Rule Service</td>
<td>• Description Rule Details Audit</td>
</tr>
<tr>
<td>Subledger Accounting</td>
<td>Journal Line Rule Service</td>
<td>• Journal Line Rule Attribute Assignment Audit</td>
</tr>
<tr>
<td>Subledger Accounting</td>
<td>Journal Entry Rule Set Service</td>
<td>• Journal Line Rule Audit</td>
</tr>
<tr>
<td>Subledger Accounting</td>
<td>Journal Entry Rule Set Service</td>
<td>• Account Rule Line Assignment Audit</td>
</tr>
</tbody>
</table>
## Maintain Common Reference Objects

<table>
<thead>
<tr>
<th>Area</th>
<th>Parent Objects</th>
<th>Child Objects</th>
</tr>
</thead>
</table>
| subledger accounting  | mapping set service                   | • mapping set entered source audit  
|                       |                                       | • mapping set flavor audit  
|                       |                                       | • mapping set value audit  
|                        | subledger accounting options service   | • subledger accounting options event class audit  
|                       |                                       | • subledger accounting system options audit  
|                       |                                       | • subledger accounting options audit  
|                        | supporting reference service          | • supporting reference audit  
|                       |                                       | • supporting reference source assignments audit  
|                        | transaction account type service      | • transaction account definition details audit  
|                       |                                       | • transaction account definition audit  
|                        | user-defined formula service          | • user-defined formula audit  
| tax                   | application tax defaults and controls | • application tax defaults and controls  
| tax                   | country tax defaults and controls     | • country tax defaults and controls  
| tax                   | party tax profile                     | • party tax profile  
| tax                   | tax formula                           | • tax formula detail  
| tax                   | tax group                             | • withholding tax group  
| tax                   | tax rate                              | • set enabled lookup type  
|                       |                                       |   • set enabled lookup values  
| tax                   | tax regime                            | • tax regime  
|                       |                                       | • default tax accounts  
| tax                   | tax reporting attributes              | • tax reporting type use  


<table>
<thead>
<tr>
<th>Area</th>
<th>Parent Objects</th>
<th>Child Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tax</td>
<td>Tax Rules</td>
<td>• Tax Rule Account Result</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tax Determining Factor Set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tax Rule Condition Set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tax Rule Result</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tax Rule Condition Details</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tax Rule Result</td>
</tr>
<tr>
<td>Tax</td>
<td>Transaction Tax</td>
<td>• Tax Code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Transaction Tax</td>
</tr>
</tbody>
</table>

**Related Topics**
- Audit History

### Impersonation Audit

Users can temporarily designate other users to impersonate their profiles and perform application tasks on their behalf. Impersonation auditing is active even when auditing is disabled for an application. At run time, the audit setup tracks and stores information about attributes, even when auditing isn't enabled for the attributes. Impersonation auditing is enabled by default so that all actions performed by the impersonator are audited. Therefore, while viewing audit history, users can retrieve the audited information, filtered by an impersonated user.

However, impersonation auditing is limited in scope and applies only to the business objects in the Manage Audit Policies task. While impersonation auditing is enabled, updates are permitted only to the business objects in the Manage Audit Policies task. The impersonator can't update business objects in other tasks. If updates to business objects in other tasks are required, the impersonation auditing must be disabled.

⚠️ **Caution:** If impersonation auditing is disabled, impersonation information won't be audited. The activities are audited as if the actual user performed them.

Impersonation auditing is controlled through the **Audit Impersonation Transaction Enabled** profile option. By default it's enabled. To disable it, set the profile value to **No**.

**Related Topics**
- Designate Proxies

### Manage Oracle Social Network Objects

#### Management of Oracle Social Network Objects

Use the Manage Oracle Social Network Objects task for managing the Oracle Social Network Objects. The integration of Oracle Social Network with applications and business processes brings key attributes from the applications to share, socialize, and update information. This helps in making better business decisions based on additional information that you obtain and analyze within your social network environment.
Use the Manage Oracle Social Network Objects page to set up and define:

- The business objects and attributes to enable
- The enablement method for social network integration with Oracle Applications Cloud

To open the Manage Oracle Social Network Objects page, use the following in the Set and Maintenance work area:

- Functional Area: Application Extensions
- Task: Manage Oracle Social Network Objects

Use Oracle Social Network to:

- Discuss projects and plans in public forums
- Maintain:
  - Membership groups
  - Activity feeds of the people you select
- Facilitate:
  - One-on-one Conversations
  - Reviews
  - Document sharing

An important aspect of managing Oracle Social Network objects is enabling business objects for integration.

**Enabling Business Objects for Integration**

A business object can't be shared within social network until a functional administrator or implementor:

- Accesses the Manage Oracle Social Network Objects page in Oracle Applications Cloud
- Enables the business object for social network integration

**Considerations for Enabling Social Networking on Objects**

You can determine whether information about a business object, such as benefit plans or sales accounts, displays in Oracle Social Network. If you enable an object for sharing, you allow users to collaborate on the object through social networking. You can choose whether all instances of an object are shared, or only at the user's discretion. You can also choose which attributes are shared, such as names, details, and who made the last update.

In addition to a wide range of predefined objects, you can share:

- Objects and attributes that you created in Application Composer
- Fields that you created in descriptive flexfields

In the Setup and Maintenance work area, use the following:

- Functional Area: Application Extensions
- Task: Manage Oracle Social Network Objects

After you click **Enable Object**, select one of the following enablement options:

- Manual
Automatic

With this option, news about all instances of the object appears on the social network, including:

- Every newly created instance
- All subsequent updates to the enabled attributes
- Deletion of any instances

No

With this option, which is the default value, no news about the object appears on the social network.

Note: When you click Disable Object, the enabled setting of the selected business object is automatically changed to No.

After you enable a business object, you must enable one or more attributes of the object. Only the enabled attributes are shared. The Status column in the Business Objects table indicates which enabled business objects don't yet have an enabled attribute. For these objects, only the following information appear on the social network:

- Internal bookkeeping information, when creating or updating an instance of the object.
- News that an instance is deleted.

Update Translations

The Update Translations process sends attribute labels and business object names to Oracle Social Network for use in the user interface.

In social network, attributes or business object labels appear in the language of your locale. If you change the locale in social network, then the attribute or business object labels appear in the updated language. However, the data appears in the language in which it was originally sent to social network. If you have previously sent an instance of the business object to social network, then the instance data isn't updated. Clicking Update Translations on the Manage Oracle Social Network Objects page sends translations for business objects with the option to enable as Manual or Automatic.

Synchronization of Business Objects

Use Synchronize on the Manage Oracle Social Network Objects page to synchronize business objects. This resends the definitions of business objects having the enablement option as Manual or Automatic to Oracle Social Network.
Use the Synchronize button at the:

- **Business Objects table level**: To resend the definitions of a selected business object to social network. This button is enabled only when you select a row for a business object with the enablement option as Manual or Automatic.

- **Manage Oracle Social Network Objects page level**: To resend the definitions of all business objects with the enablement option as Manual or Automatic to social network.

**Note**: If you had modified any business object enabled for social network and not saved your changes, then on clicking Synchronize, a warning message appears. This message informs you that you have not saved your changes, and you can select one of the following options:

- **Save and Synchronize**: To save the modified business objects, and synchronize the unmodified business objects.
- **Synchronize**: To ignore any unsaved business objects, and only synchronize the unmodified business objects.
- **Cancel**: To cancel the synchronization task.

### FAQs for Manage Oracle Social Network Objects

**What happens if I update translations?**

When you update translations, you send translations for business objects with the enablement option as Manual or Automatic to Oracle Social Network.

On updating translations, you also:

- Synchronize the newly translated text from Oracle Applications Cloud so that it can be used within social network. This means you can:
  - Install and enable a new language.
  - Take a language patch at any time.
- Send attribute labels and business object names to social network for use in its user interface.

**How can I update translations?**

Use **Update Translations** on the Manage Oracle Social Network Objects page for subsequent updates to labels and attributes.

Use the **Update Translations** button at the:

- **Business Objects table level**: To send translations for a selected business object to Oracle Social Network. This button is enabled only when you select a row for a business object with the enablement option as Manual or Automatic.

- **Manage Oracle Social Network Objects page level**: To send translations for all business objects with the enablement option as Manual or Automatic to social network.

**Note**: When you save the enablement of a business object to social network, it sends the translations as well. Hence, you need not click **Update Translations** after saving the enablement.
When do I update translations?
Run the **Update Translations** process only after you install a new language pack of Oracle Applications Cloud.
Updating translations synchronizes the newly translated text to Oracle Social Network for integration with Oracle Applications Cloud.

Note: When you save the enablement of a business object to social network, it sends the translations as well. Hence, you need not click **Update Translations** after saving the enablement.

What happens if I synchronize business objects?
When you synchronize business objects, you resend the definitions of business objects having the enablement option as **Manual** or **Automatic** to Oracle Social Network.

When do I synchronize business objects?
Run the Synchronize process after you use configuration sets to import the setup from the Manage Oracle Social Network Objects page in another environment.
You can also run the process whenever you want to synchronize the settings of business objects with social network without making changes in the Manage Oracle Social Network Objects page.

Related Topics
- Contents of the Configuration Set

Define Global Search

Overview of Global Search Setup
You have many options for setting up the **global search**, from enabling or disabling it completely to controlling what appears in the **autosuggest**. For most of the steps, use setup tasks in the Setup and Maintenance work area.

Enable Global Search
A predefined set of business objects, for example help files in Applications Help, are available for global search. These objects are grouped into categories. The search gives you results from indexed data, for example the help files with text that matches your search term.
Enabling global search includes setting up the predefined objects and categories, and starting a schedule to refresh the index so that users get the latest results.

1. Check with your help desk to make sure that objects, categories, and schedules are already set up.
2. Set the Global Search Enabled (FUSION_APPS_SEARCH_ENABLED) profile option to **Yes** at the Site level.
   - Otherwise, the **Search** field isn't available in the **global header** for any user.
   - After you set the profile option, users must sign out and sign back in to see the global search.

Note: Enabling global search is the only mandatory setup. You can skip the following setup steps, and just use the default configuration for the global search.
Define Global Search
Use the following tasks in the Application Extensions functional area to control how the global search works:

- **Manage Applications Core Global Search Configurations**: Define configurations that capture a collection of settings, and determine which configurations are enabled or used as default. Each configuration applies to the global search on all or specific applications or pages.

- **Manage Applications Core Global Search Suggestion Groups**: Define suggestion groups, which represent categories of suggestions in the autosuggest.

- **Manage Applications Core Alternate Words**: Define alternates for search terms that users enter, so that the search can also find matches based on the alternate terms.

Retain Recent Items
You can set the Recent Items Retention (FND_PURGE_RECENT_ITEMS) profile option to determine how long to retain recent items for the global search autosuggest. A page that the user opened can appear in the user’s autosuggest until the specified number of days passes. Aside from the autosuggest, this profile option also applies to the recent items tracked in **Favorites and Recent Items** in the global header. The **Favorites and Recent Items** icon is available in the global header only if your default home page layout is panel or banner.

Related Topics
- **Set Profile Option Values**

Manage Global Search Configurations

Examples of Global Search Configurations Set as Enabled or Default
Each global search configuration contains settings for the global search, and a configuration can apply to specific pages or applications. Use the Manage Global Search Configurations page to enable or disable configurations, and select the one to use as the default. The following scenarios explain which configurations actually apply to the global search, depending on the configurations that you enable or set as default.

Predefined Default
The predefined Default configuration is always enabled and set as a default. This configuration is the working default unless a custom global search configuration is also set as a default. In this scenario, you don’t enable any other configuration, so this Default configuration applies to the global search on all pages, in all applications.

Custom Default
You create a global search configuration that applies to page A and application B. Later, you set your configuration as the default. Only this configuration and the predefined Default configuration are enabled. Both are set as default.

The result is that:

- Your configuration overrides the predefined Default one and becomes the working default.
- Even though you defined your configuration to apply to page A and application B, it now actually applies to all pages and all applications. This is because your configuration is the working default, and no other configuration is enabled.
Specific Pages or Applications
You’re using either the predefined Default configuration or a custom configuration as the default. You also enable:

- **Configuration 1**: Applies to application A
- **Configuration 2**: Applies to application B and a few pages in application A

The result is that:

- Configuration 1 applies to all pages in application A, except the few pages that use configuration 2.
- Configuration 2 applies to all pages in application B, plus the few pages in application A.
- The default configuration applies to all other applications and pages.

Create Global Search Configurations
Predefined global search configurations control how the global search behaves and looks. You can’t edit these configurations, but you can duplicate them and edit the copies, or create your own from scratch.

Creating a Configuration
Follow these steps:

1. Click **Navigator > Setup and Maintenance**.
2. In the Setup and Maintenance work area, go to the following:
   - Functional Area: Application Extensions
   - Task: Manage Applications Core Global Search Configurations
3. On the Manage Applications Core Global Search Configurations page, click the **Create** icon, or select a row and click the **Duplicate** icon.
   
   **Note**: You can’t delete a configuration after you create it, but you can disable it.
4. For the short name (identifier for your configuration), enter an alphanumeric code with uppercase letters and no spaces.
5. Enter a user-friendly name and description for the configuration.
6. Select the **Default** check box to use your configuration as the default instead of the predefined Default configuration. If another custom configuration was already set as the default, then your configuration becomes the new custom default.
7. Select a product family if the configuration is for applications or pages within a specific family. Otherwise, select **Common**.
8. If you’re creating a duplicate, click **Save and Close**. To go on to the next steps and define more settings, select your configuration and click **Edit**.
9. Enter a module within the product family you selected. If you selected the Common family, then select the **Oracle Middleware Extensions for Applications** module.
10. Use the tabs to define your configuration:
   - **Autosuggest**: Determine what’s available to users in the global search autosuggest, as well as how the autosuggest looks and behaves.
   - **Search Field**: Control the Search field in the global header and in the search results dialog box.
   - **Search Results**: Enable or disable saved and recent searches, select the search categories available to users, and define settings for filters.
   - **Pages**: Indicate the applications or pages that this global search configuration applies to.
11. Save your work.

Set Up Autosuggest for Global Search

Use global search configurations to determine what’s available to users in the autosuggest. You select the suggestion groups to include in configurations. The configurations also determine how the autosuggest looks and behaves.

Prerequisite

Create a global search configuration or edit an existing one. Follow these steps:

1. In the Setup and Maintenance work area, go to the following:
   o Functional Area: Application Extensions
   o Task: Manage Applications Core Global Search Configurations

2. On the Manage Applications Core Global Search Configurations page, click the Create icon to create a global search configuration, or select an existing row and click the Edit icon to change it.

Defining the Content

To define synonyms, select suggestion groups and determine how they're displayed in the autosuggest:

1. On the page to create or edit global search configuration, click the Autosuggest tab.
2. In the Synonyms section on the Autosuggest tab, select Enable Synonyms to let users search using synonyms.
3. Optionally, enter the following:
   o **Synonym Context Code**: A logical grouping such as a product family, navigation group, or work area that determines the subset of synonyms to search on. For example, if you enter Academics, every time the user enters a term and clicks the Search icon, the search runs on all terms related to Academics and their synonyms.
   o **Synonym Object Type**: A business object name that would determine the subset of synonyms to search on. For example, if you enter Curriculum, every time the user enters a term and clicks the Search icon, the search runs on all terms related to Curriculum and their synonyms.
4. Click Manage Synonyms.
5. To add search terms and create synonyms for them, on the Manage Synonyms page:
   a. Click the New icon and do the following:
      • Enter a short name in uppercase, with no spaces. This name is required as an identifier.
      • Select a module name, which is required to map the data with a specific product or area. There are no restrictions on where the synonyms can be used.
      • Optionally, enter a context code and object type. This information can be used to ensure that certain synonyms are only used when a global search configuration having the same values is in effect.
      • Enter the primary synonym, which is the main term or business object to which other synonyms are to be matched.
      • Enter at least one synonym that matches the primary synonym.
   b. Click Save and Close.
6. Select the Enable personalization of search groups check box to allow users to override your configuration. Users can hide, show, and reorder suggestion groups for their autosuggest.
7. In the Suggestion Group section on the Autosuggest tab, move the groups you want to include into the Selected Groups pane. Use the Move, Move All, Remove, and Remove All icons to move the groups.
   The Enabled column in the Available Groups pane indicates if the group is defined (on the Manage Suggestion Groups page) to be displayed by default or not in the autosuggest.
In the **Enabled** column in the Selected Groups pane, select one of the following values:

- **Inherit**: In the autosuggest, the group is displayed or hidden by default depending on what’s defined for the group.
- **Yes**: The group is displayed by default, no matter what’s defined for the group.
- **No**: The group is hidden by default, no matter what’s defined for the group.

**Note:** The **Displayed by Default** column shows the resulting behavior in the autosuggest, based on what you select in the **Enabled** column.

Using the **Move to Top**, **Move Up**, **Move Down**, and **Move to Bottom** icons, order the selected groups as you want them to appear in the autosuggest.

**Tip:** Click **Manage Suggestion Groups** at any time to edit or create suggestion groups. When you return to the Autosuggest tab, click **Refresh** to reflect the changes you made to suggestion groups.

### Defining the Appearance

Optionally define settings in the Appearance section on the Autosuggest tab:

- **Show Suggestion Group Headings**: Select this option to display suggestion group headings (text and icon) in the autosuggest. Even if you do so, if a group is defined on the Manage Suggestion Groups page to not show headings, then its heading won’t be displayed.
- **Show Icons**: Select this option to display icons next to suggestions in the autosuggest.
- **No Suggestions Message**: Enter the message that appears when no suggestions match the user’s search term. If you leave this field blank, then no autosuggest or anything at all appears when there are no matches.

### Defining the Behavior

Optionally define settings in the Behavior section on the Autosuggest tab:

- **Show Top Suggestions**: Enable this option to display suggestions in the autosuggest as soon as the user clicks in the search field, even without entering a search term. For example, the last few pages the user opened would appear as suggestions in the Recent Items group.
- **Minimum Characters for Autosuggest**: Enter the number of characters that users must enter in the search field before matching suggestions appear in the autosuggest.
- **Maximum Number of Suggestions**: Enter the maximum number of suggestions to be displayed across all suggestion groups. This total is distributed as equally as possible among the groups.

### Considerations for Disabling Saved Searches and Recent Searches for the Global Search

Global search configurations determine if saved searches and recent searches are enabled in the **global search**. Consider the following points when you disable either. In the Create or Edit Global Search Configuration page, open the Search Results tab and use the Saved and Recent Searches section.

#### Disabling Saved Searches

If you disable saved searches:

- You disable the Save button in the search results dialog box, so users can't create or edit saved searches for global search.
- You’re not disabling the Saved Searches **suggestion group**. Users can still see any applicable saved searches in the global search **autosuggest**.
Disabling Recent Searches
If you disable recent searches:

- The application isn’t saving recent searches.
- You’re not disabling the Recent Searches suggestion group. Users can still see in the autosuggest any applicable searches that they recently ran before you disabled recent searches.

Set Up Filters for Global Search
To determine how users can filter their search results, define the appropriate settings when you create or edit global search configurations. You can set up search categories so that users can limit the scope of their search to begin with, or refine their search results.

Prerequisite
Create a global search configuration or edit an existing one. Follow these steps:

1. In the Setup and Maintenance work area, go to the following:
   - Functional Area: Application Extensions
   - Task: Manage Applications Core Global Search Configurations
2. Click the Create icon to create a global search configuration, or select an existing one and click the Edit icon to change it.

Setting Up Categories to Narrow Search Scope
To let users select the categories to search on before running the search:

1. On the page to create or edit global search configuration, click the Search Results tab.
2. In the Saved and Recent Searches section, select the check boxes to enable saved searches and recent searches.
3. In the Filters section, enable personalization of search categories.
4. In the Search Categories section, select the categories that users can search on. If you don’t select any, then every category is available to users.
5. For each of your selected categories, select Yes or No in the Enabled column to have it enabled or disabled by default.

Setting Up Categories as Search Result Filters
To let users filter search results based on category:

1. In the Filters section, select a filter display style so that the list of all available category names are displayed:
   - Inline: In the Filters pane in the search results
   - LOV: In a Categories dialog box that users can open from the Filters pane
2. Select the check boxes to show subcategories, facets, or both. Categories are always displayed. Subcategories are an additional level of filters that appear after categories, and facets are a level after categories.
3. In the Search Categories section, select the categories to use as filters. This is the same set of categories to be used for personalization. If you don’t select any, then every category is available to users.

Setting Other Options for Filters
You can also use the Filters section to:

- **Show hit counts:** Show the number of search results that match each filter value
• Enable clear all filters: Allow users to clear all filters with one click of a button
• Show applied filters: Show all the filters that have been applied

In the Last Updated Date Filters section, select the criteria to use as filters, based on the last update date. If you don’t select any, then every date filter is available to users.

In the Appearance section, set the Show Icons option Off if you don’t want to display the icons.

Specify the Pages or Applications That a Global Search Configuration Applies To
As part of defining your global search configuration, you can specify the pages or applications (or both) that your configuration applies to. If you want your configuration to apply to all pages in all applications, then skip these steps.

Before You Start
If you want the global search configuration to apply to specific applications, you need to find the application short name.

1. Click Navigator > Setup and Maintenance.
2. In the Setup and Maintenance work area, go to the following:
   o Functional Area: Application Extensions
   o Task: Manage Taxonomy Hierarchy
3. On the Manage Taxonomy Hierarchy page, expand the Oracle Fusion node.
4. Select the row (with the Application module type) for your application, and click Edit Module.
5. In the Application Details section, see the Application Short Name column and note down the value to use as the application short name.

Add Pages or Applications
Follow these steps:

1. On the Create or Edit Global Search Configurations page, open the Pages tab.
2. Click the Create icon.
3. In the View Type column, indicate if the configuration applies to a specific page or application.
4. Enter a view ID to identify the page or application:
   o Tip: You can use % as a wildcard for the page or application value, such as Example% for all pages that start with Example.
   o Page: Enter the last part of the URL you get when you open that page. For example, enter ExamplePage from the URL http://exampleServer/homePage/faces/ExamplePage.
   o Application: Enter the application short name with a wildcard at the end, for example HomePageApp%.
5. Add more pages or applications as needed.

Manage Suggestion Groups

Suggestion Groups for Global Search
A suggestion group is a category of suggestions that users see in the autosuggest for the global search. For example, if the user enters Report in the Search field, then the Navigator suggestion group in the autosuggest shows any Navigator menu names with Report.
Managing Suggestion Groups

Each suggestion group can have a heading, for example Recent Items, as well as an icon that appears before the heading text. The icon helps users identify what the group is all about.

You can manage suggestion groups to:

- Show or hide the group by default in the autosuggest
- Enter the heading text
- Define if the heading text appears in the autosuggest or not
- Identify the image file to use as the icon

**Tip:** You can copy predefined suggestion groups and edit them to create your own versions.

To manage suggestion groups, open the Setup and Maintenance work area and use either the:

- Manage Suggestion Groups task
- Autosuggest tab when you create or edit global search configurations using the Manage Global Search Configurations task

Using in Global Search Configurations

Just because a suggestion group is defined to be displayed by default doesn't necessarily mean that it in fact appears in the autosuggest. Global search configurations determine which groups are included for the autosuggest, whether icons appear for the headings, and so on.

You can use a suggestion group in many or all global search configurations. And ultimately, if personalizing suggestion groups is enabled, then users can show, hide, and reorder the suggestion groups included in the global search configuration.

Change Heading Text and Icons for Suggestion Groups

This example shows how to change the icon and text for a suggestion group heading in the global search autosuggest. In this example, you start out with the predefined Default global search configuration enabled, and no custom configurations.

The following table summarizes key decisions for this scenario.

<table>
<thead>
<tr>
<th>Decisions to Consider</th>
<th>In This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Which predefined suggestion group do you want to change?</td>
<td>Recent Items</td>
</tr>
<tr>
<td>What do you want the new heading text to be?</td>
<td>Recently Visited Pages</td>
</tr>
<tr>
<td>Which image do you want to use as the icon?</td>
<td>A .png file (16 by 16 pixels) that's used on your company Web site</td>
</tr>
<tr>
<td>Do you want the new suggestion group to appear in the autosuggest by default?</td>
<td>Yes</td>
</tr>
<tr>
<td>Which pages should the changes apply to?</td>
<td>All pages</td>
</tr>
</tbody>
</table>
Decisions to Consider | In This Example
--- | ---

To use a new icon and heading text for the suggestion group:

- Make a copy of the predefined Recent Items suggestion group.
- Update global search configurations to use your suggestion group.

**Duplicate the Predefined Suggestion Group**

1. Click **Navigator > Setup and Maintenance**.
2. In the Setup and Maintenance work area, go to the following:
   - Functional Area: Application Extensions
   - Task: Manage Applications Core Global Search Suggestion Groups
3. On the Manage Applications Core Global Search Suggestion Groups page, select the Recent Items group and click **Duplicate**.
4. In the new row, enter `RECENTPAGES` as the short name.
5. Change the display name to **Recent Pages**.
6. Change the description to **Version of Recent Items with revised heading and icon**.
7. With your new row still selected, click **Edit**.
8. In the Heading section, enter **Recently Visited Pages** in the **Text** field.
9. In the **Icon** field, enter the full URL to your .png file.
10. Click **Save and Close**.

**Update Global Search Configurations**

1. Back in the Setup and Maintenance work area, go to the following:
   - Functional area: Application Extensions
   - Task: Manage Applications Core Global Search Configurations
2. Select the Default configuration and click **Duplicate**.
3. Fill out the row for your new configuration, selecting the **Default** check box.
4. With the row still selected, click the **Edit** icon.
5. In the Autosuggest tab, click the **Refresh** button in the Suggestion Group section if you don’t see your suggestion group.
6. Move the Recent Pages group into the **Selected Groups** list, and move the Recent Items group out.
7. In the **Enabled** column for the Recent Pages group, select **Yes**.
8. In the Appearance section, make sure that headings are set to be displayed.
9. Click **Save and Close**.

**Considerations for Managing Alternate Words for Global Search**

Use the Manage Applications Core Alternate Words task in the Setup and Maintenance work area to maintain a list of search terms that users might use for the **global search**. For each user keyword, define a possible alternate to also search on. Consider various reasons for defining these word pairs, and also decide whether to automatically search on both terms or not.
Reasons for Word Pairs
This table provides some reasons and examples for managing alternate words.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>What You Enter</th>
<th>User Input Keyword Example</th>
<th>Alternate Keyword Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct user typos</td>
<td>A possible typo as the user input keyword, and the correctly spelled term as the alternate keyword</td>
<td>Oracel</td>
<td>Oracle</td>
</tr>
<tr>
<td>Account for abbreviations and acronyms</td>
<td>An abbreviation or acronym and what it stands for</td>
<td>Bl</td>
<td>business intelligence</td>
</tr>
<tr>
<td>Account for common variations in spelling</td>
<td>Two different ways to spell the same term</td>
<td>email</td>
<td>e-mail</td>
</tr>
<tr>
<td>Enable matches on synonyms</td>
<td>A pair of terms that are functionally similar</td>
<td>hyperlink</td>
<td>link</td>
</tr>
<tr>
<td>Help new users who are not familiar with what things are called in the application</td>
<td>The term that your users might search on as the user input keyword, and the equivalent in the application as the alternate keyword</td>
<td>concurrent program</td>
<td>scheduled process</td>
</tr>
</tbody>
</table>

**Note:** You don't have to account for plurals or case sensitivity. For example, if you have email as a user input word, you don't have to also add Email or emails as an input word.

Automatically Search Both Terms
For each pair of terms, use the Automatically Search Both check box to determine what happens when the user enters the input word and starts the search:

- **Yes:** The search runs and displays results based on both terms.

  Before you select this check box, carefully consider possible impact. For example, would users get a lot of unnecessary search results, making it harder to find what they want?

- **No:** The user sees a message and can decide to continue searching on just the input term, or to search on just the alternate term instead.
3 Define Currencies and Currency Rates

Manage Currencies

Considerations for Defining Currencies

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

Currency Codes
You can’t change a currency code after you enable the currency, even if you later disable that currency.

Date Ranges
You can enter transactions denominated in the currency only for the dates within the specified range. If you don't enter a start date, then the currency is valid immediately. If you don’t enter an end date, then the currency is valid indefinitely.

Symbols
Some applications support displaying currency symbols. You may enter the symbol associated with a currency so that it appears along with the amount.

Related Topics
- What's the difference between precision, extended precision, and minimum accountable unit for a currency
- What's a statistical unit currency type

Euro Currency Derivation

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. If you have to use a different currency for Euro, you can disable the predefined 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 don’t 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 must 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.
Manage Conversion Rate Types

Guidelines for Creating Conversion Rate Types

Maintain different conversion rates between currencies for the same period using conversion rate types. The following conversion rate types are predefined:

- Spot
- Corporate
- User
- Fixed

You can use different rate types for different business needs. During journal entry, the conversion rate is provided automatically based on the selected conversion rate type and currency, unless the rate type is User. For User rate types, you must enter a conversion rate. You can define additional rate types as needed. Set your most frequently used rate type as the default. Conversion rate types can't be deleted.

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

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

- Convert foreign currency journal amounts to ledger currency equivalents.
- Convert journal amounts from source ledgers to reporting currencies or secondary ledgers.
- Run revaluation or translation processes.

When creating conversion rates, decide whether to:

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

Enforce Inverse Relationships

The Enforce Inverse Relationship option indicates whether to enforce the automatic calculation of inverse conversion rates when defining daily rates. The following table describes the impact of selecting or not selecting the option.

<table>
<thead>
<tr>
<th>Action</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected</td>
<td>When you enter a daily rate to convert currency A to currency B, the inverse rate of currency B to currency A is automatically calculated and entered in the adjacent column. If either rate is changed, the application automatically recalculates the other rate.</td>
</tr>
</tbody>
</table>
### Select Pivot Currencies
Select a pivot currency that is commonly used in your currency conversions. A pivot currency is the central currency that interacts with contra currencies. For example, you set up a daily rate between the US dollar (USD) and the Euro currency (EUR) and another between the USD and the Canadian dollar (CAD). USD is the pivot currency in creating a rate between EUR and CAD. EUR and CAD are the contra currencies. Select the pivot currency from the list of values which contains those currencies that are enabled, effective, and not a statistical (STAT) currency. The description of the pivot currency is populated automatically based on the currency definition.

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

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

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

For example, if you have daily rates defined for the pivot currency, USD to the contra currency, EUR, and USD to another contra currency, CAD, the application automatically creates the rates between EUR to CAD and CAD to EUR. You don't have to manually define the EUR to CAD and CAD to EUR rates.

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

### Maintain Cross Rate Rules
Define or update your cross rate rules at any time by adding or removing contra currency assignments. Add a contra currency to a cross rate rule and run the Daily Rates Import and Calculation process to generate the new rates. If you remove a cross rate rule or a contra currency from a rule, any cross rates generated previously for that contra currency remain unless you manually delete them. Changes to the rule aren't retroactive and don't affect previously stored cross rates. The Cross Rate process generates as many rates as possible and skips currencies where one component of the set is missing.
Note: With a defined web service that extracts daily currency conversion rates from external services, for example Reuters, currency conversion rates are automatically updated for the daily rates and all cross currency relationships.

Related Topics
- What's the difference between calendar and fiscal period naming

Examples of Using Conversion Rate Types in a Journal

The four predefined conversion rate types are:
- Spot
- Corporate
- User
- Fixed

Scenario

You are the general ledger accountant for Vision US Inc. You are entering a journal entry to capture three transactions that were transacted in three different foreign currencies.

- Canadian Dollar CAD: A stable currency
- Mexican Peso MXP: A fluctuating currency
- Hong Kong Dollar HKD: An infrequently used currency

You enter two journal lines with accounts and amounts for each foreign currency transaction. Based on your company procedures, you select the rate type to populate the rate for Corporate and Spot rate types from your daily rates table. You manually enter the current rate for the User rate type.

The following table lists the currency, the rate type that you select, and the reasons for the rate type selection.

<table>
<thead>
<tr>
<th>Selected Currency</th>
<th>Selected Rate Type</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAD</td>
<td>Corporate</td>
<td>Entered a periodic type of transaction. Your company has established a daily rate to use for the entire month across divisions for all transactions in Canadian dollars, a stable currency that fluctuates only slightly over the month.</td>
</tr>
<tr>
<td>MXP</td>
<td>Spot</td>
<td>Entered a periodic type of transaction. Your company enters daily rates each day for the Mexican peso because the currency is unstable and fluctuates.</td>
</tr>
<tr>
<td>HKD</td>
<td>User</td>
<td>Entered a one time transaction. Your company doesn't maintain daily rates for Hong Kong dollars.</td>
</tr>
</tbody>
</table>
Your company doesn’t currently use the **Fixed** rate type. From January 1, 1999, the conversion rate of the French franc FRF against the Euro EUR was a fixed rate of 1 EUR to 6.55957 FRF. Your French operations were started in 2007, so you maintain all your French business records in the Euro.

**FAQs for Manage Conversion Rate Types**

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

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

- **Spot**: For currencies with fluctuating conversion rates, or when exact currency conversion is needed.
- **Corporate**: For setting a standard rate across your organization for a stable currency.
- **User**: For infrequent entries where daily rates for the entered foreign currency aren’t set up.
- **Fixed**: For rates where the conversion is constant between two currencies.

If you have infrequent foreign currency transactions, the User rate type can simplify currency maintenance. The User rate type can also provide an accurate conversion rate on the date of the transaction.
4 Define 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 that describe its operations and provide a basis for reporting.

- Legal
- Managerial
- Functional

In Oracle Fusion, these structures are implemented using the chart of accounts and organization hierarchies. 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
- Departments

Align these structures with your strategic objectives.
Legal Structure

The figure illustrates a typical group of legal entities, operating various business and functional organizations. Your ability to buy and sell, own, and employ comes from your charter in the legal system. A corporation is:

- A distinct legal entity from its owners and managers.
Many other kinds of legal entities exist, 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 the entity 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 within the laws of each country in which their enterprise operates.

The figure illustrates:

- 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 contributes to 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 its 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 don't have to be reflected directly in the legal structure of the enterprise. The management structure can include divisions, subdivisions, lines of business, strategic business units, profit, and cost centers. In the figure, the management structure is shown on the Business Axis. In Oracle Fusion Applications, the management structure is implemented using divisions and business units as well as being reflected in the chart of accounts.

### 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. 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 and selling, general, and administrative 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

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 includes:

- Set Up Enterprise Structures business process, which consists of implementation activities that span many product families.
- Information Technology, a second Business Process Model which contains the Set Up Information Technology Management business process.
- Define Reference Data Sharing, which 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 to implement.
The following figure and tablet describe the Business Process Model structures and activities.

The table describes each BPM activity.

<table>
<thead>
<tr>
<th>BPM Activities</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define Enterprise</td>
<td>Define the enterprise to get the name of the deploying enterprise and the location of the headquarters.</td>
</tr>
<tr>
<td>Define Enterprise Structures</td>
<td>Define enterprise structures to represent an organization with one or more legal entities. Define organizations to represent each area of business within the enterprise.</td>
</tr>
<tr>
<td>BPM Activities</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Define Legal Jurisdictions and Authorities</td>
<td>Define information for governing bodies that operate within a jurisdiction.</td>
</tr>
<tr>
<td>Define Legal Entities</td>
<td>Define legal entities and legal reporting units for business activities handled by the Oracle Fusion Applications.</td>
</tr>
<tr>
<td>Define Business Units</td>
<td>Define business units of an enterprise to perform 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 has a manager, strategic objectives, a level of autonomy, and responsibility for its profit and loss.</td>
</tr>
<tr>
<td>Define Financial Reporting Structures</td>
<td>Define financial reporting structures, including organization structures, charts of accounts, organizational hierarchies, calendars, currencies and rates, ledgers, and document sequences which are used in organizing the financial data of a company.</td>
</tr>
<tr>
<td>Define Chart of Accounts</td>
<td>Define chart of accounts including hierarchies and values to enable tracking of financial transactions and reporting at legal entity, cost center, account, and other segment levels.</td>
</tr>
<tr>
<td>Define Ledgers</td>
<td>Define the primary accounting ledger and any secondary ledgers that provide an alternative accounting representation of the financial data.</td>
</tr>
<tr>
<td>Define Accounting Configurations</td>
<td>Define the accounting configuration that serves as a framework for how financial records are maintained for an organization.</td>
</tr>
<tr>
<td>Define Facilities</td>
<td>Define your manufacturing and storage facilities as Inventory Organizations if Oracle Fusion tracks inventory balances there and Item Organizations if Oracle Fusion only tracks the items used in the facility but not the balances.</td>
</tr>
<tr>
<td>Define Reference Data Sharing</td>
<td>Define how reference data in the applications is partitioned and shared.</td>
</tr>
</tbody>
</table>

**Note:** Some product-specific implementation activities aren’t listed here and depend on the applications you’re implementing. For example, you can implement Define Enterprise Structures for Human Capital Management, Project Management, and Sales Management.

### Guidelines for Configuring Global Enterprises

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. The following are some questions and points to consider as you design your global enterprise structure in Oracle Fusion.

- Enterprise Configuration
- Business Unit Management
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?

Model Your Enterprise Management Structure

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 US dollars. Your product line includes all the components to build and maintain air quality monitoring applications 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 initial costs of these applications.

Analysis

The following are elements you must 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 want 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 do you use 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 units in which to perform all these functions? How is your shared service center configured?

Global Enterprise Structure Model

The following figure and table summarize the model that your committee has designed and uses numeric 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 application components, installations, and maintenance product lines across the enterprise
- All UK general and administrative costs processed at the UK headquarters
- US Systems’ general and administrative costs processed at US Corporate headquarters
- US Financial Services maintains its own payables and receivables departments
In this chart, the green globe stands for required and gold globe stands for optional setup. The following statements expand on the data in the chart.

- The enterprise is required because it serves as an umbrella for the entire implementation. All organizations are created within an enterprise.
- Legal entities are also required. They can be optionally mapped to balancing segment values or represented by ledgers. Mapping balancing segment values to legal entities is required if you plan to use the intercompany functionality. The InFusion Corporation is a legal entity but isn’t discussed in this example.
- At least one ledger is required in an implementation in which you record your accounting transactions.
- Business units are also required 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 required because they track your employees.
- Optionally, add an item master organization and inventory organizations if you're tracking your inventory transactions in Oracle Fusion Applications.
Note: Some Oracle Fusion Human Capital Management implementations don't require recording accounting transactions and therefore, don't require a ledger.

Essbase Character and Word Limitations

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

The following table shows how the General Ledger configuration maps to Essbase.

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

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

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

### Restricted Characters

The following table lists the characters that are restricted and can't be used in dimension, member, or alias names.

<table>
<thead>
<tr>
<th>Character</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
<td>ampersand</td>
</tr>
<tr>
<td>@</td>
<td>at sign</td>
</tr>
<tr>
<td>\</td>
<td>backslash</td>
</tr>
<tr>
<td>()</td>
<td>brace</td>
</tr>
<tr>
<td>,</td>
<td>comma</td>
</tr>
<tr>
<td>-</td>
<td>dash, hyphen, or minus sign</td>
</tr>
<tr>
<td></td>
<td>For the accounting calendar period names, you can use a hyphen or an underscore in the middle of an accounting calendar period name. For example: Jan-15 or Adj_Dec-15 can be used successfully.</td>
</tr>
<tr>
<td>=</td>
<td>equal sign</td>
</tr>
<tr>
<td>&lt;</td>
<td>less than sign</td>
</tr>
<tr>
<td>()</td>
<td>parentheses</td>
</tr>
<tr>
<td>.</td>
<td>period</td>
</tr>
<tr>
<td>+</td>
<td>plus sign</td>
</tr>
<tr>
<td>'</td>
<td>single quotation mark</td>
</tr>
<tr>
<td>_</td>
<td>underscore</td>
</tr>
<tr>
<td></td>
<td>For the accounting calendar period names, you can use a hyphen or an underscore in the middle of an accounting calendar period name. For example: Jan-15 or Adj_Dec-15 can be used successfully.</td>
</tr>
</tbody>
</table>
Other Restrictions

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

The following table lists additional words that should not be used.

<table>
<thead>
<tr>
<th>List 1</th>
<th>List 2</th>
<th>List 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>AND</td>
<td>ASSIGN</td>
</tr>
<tr>
<td>AVERAGE</td>
<td>CALC</td>
<td>CALCMBR</td>
</tr>
<tr>
<td>COPYFORWARD</td>
<td>CROSSDIM</td>
<td>CURMBRNAME</td>
</tr>
<tr>
<td>DIM</td>
<td>DIMNAME</td>
<td>DIV</td>
</tr>
<tr>
<td>DYNAMIC</td>
<td>EMPTYTERM</td>
<td>EQ</td>
</tr>
<tr>
<td>EQOP</td>
<td>EXCEPT</td>
<td>EXP</td>
</tr>
<tr>
<td>EXPERROR</td>
<td>FLOAT</td>
<td>FUNCTION</td>
</tr>
<tr>
<td>GE</td>
<td>GEN</td>
<td>GENRANGE</td>
</tr>
<tr>
<td>GROUP</td>
<td>GT</td>
<td>ID</td>
</tr>
<tr>
<td>IDERROR</td>
<td>INTEGER</td>
<td>LE</td>
</tr>
</tbody>
</table>
Define Initial Configuration with the Enterprise Structures Configurator

Establishing Enterprise Structures Using the Enterprise Structures Configurator: Explained

The Enterprise Structures Configurator is an interview-based tool that guides you through the process of setting up a basic enterprise structure. By answering questions about your enterprise, the tool creates a structure of divisions, legal entities, business units, and reference data sets that reflects your enterprise structure. After you create your enterprise structure, you also follow a guided process to determine whether 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 don't select this feature, then you must set up your enterprise structure using individual tasks provided elsewhere in the offerings, and you can't create multiple configurations to compare different scenarios.

**Establish Enterprise Structures**

To define your enterprise structures, use the guided flow within the Establish Enterprise Structures task to enter basic information about your enterprise, such as the primary industry. 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 enough information 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, you are prompted to set up a descriptive flexfield structure for jobs, and for positions if applicable. Descriptive flexfields enable you to get more information when you create jobs and positions.

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

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

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

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

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

Design an Enterprise Configuration
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 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 company revenue of 120 million US dollars. Outside the US, InFusion employs 200 people and has revenue of 60 million US dollars.
Analysis

InFusion requires three divisions.

- The US division covers the US locations.
- The Europe division covers UK and France.
- Saudi Arabia and the UAE are 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, 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 has general managers responsible for business units within each country. Those business units may share reference data. Some reference data can be defined within a reference data set that multiple business units may subscribe to. Business units are also required for financial purposes. Financial transactions are always processed within a business unit.

Resulting Enterprise Configuration

Based on this analysis, InFusion requires an enterprise with multiple divisions, ledgers, legal employers, payroll statutory units, tax reporting units, legislative data groups, departments, cost centers, and business units.
This figure illustrates the enterprise configuration that results from the analysis of InFusion Corporation.

**Divisions**

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 don’t reflect directly the legal structure of the enterprise. The management entities and structure can include:

- Divisions and subdivisions
- Lines of business
- Other strategic business units
- 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 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 achieving business goals including profits. 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 generally reports to a corporate executive.

By definition a division can be represented in the chart of accounts. Companies can use product lines, brands, or geographies as their divisions: their choice represents the primary organizing principle of the enterprise.

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.

How You Create Legal Entities in the Enterprise Structures Configurator

Use the Enterprise Structures Configurator, to 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, 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 its enterprise structure. The corporation has 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.

![Enterprise Structure Diagram]

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

<table>
<thead>
<tr>
<th>Country</th>
<th>Enterprise</th>
<th>InFusion Lighting</th>
<th>InFusion Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Japan</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>US</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>UK</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>India</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Based on the selections made in the preceding table, the ESC creates the following four legal entities:

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

Creating Legal Entities Using a Spreadsheet

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

Related Topics

- Examples of HCM Organization Models
- Guidelines for Using Desktop Integrated Excel Workbooks

Considerations for Creating Business Units in the Enterprise Structures Configurator

Business units are used within Oracle Fusion applications for management reporting, processing of transactions, and security of transactional data. Using the Enterprise Structures Configurator, 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 one of two levels:

- Business function level, such as Sales, Consulting, Product Development, and so on.
- 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 application displays a table listing both components. You select the check boxes at the intersections of the two components.

The business units listed by the application are suggestions only, and are meant to simplify the process to create business units. You aren’t required to select all of the business units suggested. When you navigate to the next page in the ESC guided flow, the Manage Business Units page, you can’t delete any of the business units 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 its enterprise structure. InFusion has identified two divisions, one for Lighting, and one for Security. They operate in four countries: US, UK, Japan, and India, and they have created a legal entity for each of the countries. The sales and marketing functions are based in both India and Japan, while the US and the UK have only the sales function.

This figure illustrates InFusion Corporation’s enterprise structure.

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

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

**Manually Creating Business Units**

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

Reference data sharing enables you to group set-enabled reference data such as jobs or grades to share the data across different parts of the organization. Sets also enable you to filter reference data at the transaction level so that only data assigned to certain sets is available to be selected. 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 assigned to the Common Set is always available, in addition to the reference data assigned to the set that corresponds to the business unit on the transaction.

Other types of reference data can be specific to certain business units, so you can 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 to use 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 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. They 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
The following table describes the default set assignment and the set assignment overrides for each business unit in InFusion:

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

When setting up grades, departments, and locations for the business units, InFusion assigns the data to the default set for each business unit. When setting up jobs, they assign the Jobs set and 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 can select data from the set that corresponds to the business unit they enter on the transaction, and any data assigned to the Common Set. For example, for transactions for the Marketing_Japan business unit, grades, locations, and departments from the Mktg_Japan_Set is available to select, as well as from the Common Set.

When using jobs at the transaction level, users can select jobs from the Jobs set and from the Common Set when they enter a sales business unit on the transaction. For example, when a manager hires an employee for the Sales_India business unit, the list of jobs is filtered to show jobs from the Jobs and Common sets.

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 the:

- Common set
- 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 includes grade in the:

- Common set
- Set that is assigned to grades for the business unit in which you creating the assignment

Examples of Positions

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

Retail Industry
ABC Corporation has high turnovers. It loses approximately 5% of its cashiers monthly. The job of the cashier includes three positions: front line cashier, service desk cashier, and layaway cashier. Each job is cross-trained to take over
another cashier’s 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 Corporation must replace each cashier lost to turnover. Since turnover is high in retail it’s better for this industry to use positions.

Note the following:

- An automatic vacancy is created when an employee terminates employment.
- The position exists even when there are no holders. Having the position continue to exist 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 the position is empty.
- You don’t have to reassign these employees to another manager or supervisor. The replacement manager is assigned to the existing position.

Also, an added advantage to using Positions is when you hire somebody new, many of the attributes are inherited from the position. This speeds up the hiring process.

This figure illustrates the retail position setup.

Health Care Industry

Health care is an industry that must 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.
The hospital has a structured headcount and detailed budgeting. For example, a specific number of surgeons, nurses, and interns of various types are needed. These positions must be filled in order for the hospital to run smoothly. Use jobs and positions when you apply detailed headcount rules.

This figure illustrates the hospital position setup.

---

**Examples of Jobs**

*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 resigned from the company.
- The director decides to redirect the headcount 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 headcount, 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

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 to identify additional details about the job, such as the nature of the work that is performed or the relative skill level required. 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. A position belongs 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, consider whether 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. 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 can't set up multiple configurations and compare different scenarios. Using the Enterprise Structures Configurator is the recommended process for setting up your enterprise structures.
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 these levels:

- Ultimate holding company as the top level
- **Divisions** and **country holding companies** as the second level
- **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.
- Jobs are assigned to the default set, Vision 3 SET.

Define Reference Data Sharing

Reference Data Sharing

**Reference data** sharing facilitates sharing of configuration data such as jobs and payment terms, across organizational divisions or business units. You define reference data sets and determine how common data is shared or partitioned across business entities to avoid duplication and reduce maintenance effort. 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.

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 can’t 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 different business units setting up and using the same grades, XYZ Corporation decides to create a set called Grades, which contains the grades. All business units in the organization have the Grades set so that the grades can be shared and used.

**Note:** For specific information about configuring reference data sharing for a particular object or product, refer to the relevant product documentation.
Reference Data Sets

Reference data sets are logical groups of reference data that various transactional entities can use depending on the business context. You can get started using either the common reference data set or the enterprise set depending on your implementation requirement. You can also 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 only some aspects of corporate policy should affect all business units. The remaining aspects are at the discretion of the business unit manager to implement. This enables your enterprise to balance autonomy and control for each business unit. For example, your enterprise holds business unit managers accountable for their profit and loss, but manages working capital requirements at a corporate level. Then, you can let managers define their own sales methods, but define payment terms centrally. As a result, each business unit has its own reference data set for sales methods and one central reference data set for payment terms assigned to all business units.

Partitioning

Partitioning reference data and creating data sets provide you the flexibility to handle the reference data to fulfill your business requirements. You can share modular information and data processing options among business units with ease. You can create separate sets and subsets for each business unit. Alternatively, you can create common sets or subsets to enable sharing reference data between several business units, without duplicating the reference data.

The following figure illustrates the reference data sharing method. 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.
Related Topics

- Define Default Reference Data Sets

Reference Data Sets and Sharing Methods

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. As a result, there is a reduction in the administrative burden and the time to implement new business units. For example, you can share sales methods, or transaction types across business units. You may also share certain 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 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's being shared. For example, assign separate sets for payment terms, transaction types, and sales methods to your business units.

Your enterprise can determine that certain aspects of your corporate policy can affect all business units. The remaining aspects are at the discretion of the business unit manager to implement. This allows your enterprise to balance autonomy and control for each business unit. For example, your enterprise holds business unit managers accountable for their profit and loss, but manages working capital requirements at a corporate level. In such a case, you can let managers define their own sales methods, but define payment terms centrally. In this example:

- Each business unit has its own reference data set for sales methods.
- One central reference data set for payment terms is 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 set up 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

Variations exist 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. This method is 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. This method is 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's available to all the business units. You need not
explicitly assign 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 the following:

- Transaction types from the set assigned to the business unit.
- Transaction types assigned to the common set that's 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 assign Net 15 to a set specific only to your business unit. At transaction entry, the list of values for payment terms consists of only the set that's 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. Also update the data set going forward as you create new reference data items.

## Assignment of Reference Data Sets to Reference Objects

You can assign the reference data sets to reference objects using 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 the 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 using a business context setting called the determinant type. A determinant type is the point of reference used in the data assignment process. The following table lists the determinant types used in the reference data assignment.

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

The determinant (also called determinant value) is a value that corresponds to the selected determinant type. The determinant is one of the criteria for selecting the appropriate reference data set.

Reference Groups

A transactional entity may have multiple reference entities (generally considered to be setup data). However, all reference entities are treated alike because of similarity in implementing business policies and legal rules. Such reference entities in your application are grouped into logical units called reference groups. For example, all tables and views that define Sales Order Type details might be a part of the same reference group. Reference groups are predefined in the reference groups table.

Items and Supplier Site Reference Data Sharing

Some products, such as items and supplier sites, required special logic for reference data sharing and have implemented their own domain-specific ways for sharing data.

Items

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

Suppliers Sites

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

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

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

**What reference data objects can be shared across business units?**

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

<table>
<thead>
<tr>
<th>Application Name</th>
<th>Reference Data Object</th>
<th>Method of Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trading Community Model</td>
<td>Customer Account Relationship</td>
<td>Assignment to one set only, no common values allowed</td>
</tr>
<tr>
<td>Trading Community Model</td>
<td>Customer Account Site</td>
<td>Assignment to one set only, no common values allowed</td>
</tr>
<tr>
<td>Trading Community Model</td>
<td>Salesperson</td>
<td>Assignment to one set only, no common values allowed</td>
</tr>
<tr>
<td>Opportunity Management</td>
<td>Sales Method Group</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Work Management</td>
<td>Assessment Templates</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Enterprise Contracts</td>
<td>Contract Types</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Sales</td>
<td>Sales Method</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Common Components</td>
<td>Activity Templates</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Payables</td>
<td>Payment Terms</td>
<td>Assignment to multiple sets, no common values allowed</td>
</tr>
<tr>
<td>Receivables</td>
<td>Accounting Rules</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Receivables</td>
<td>Aging Buckets</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Receivables</td>
<td>Auto Cash Rules</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Application Name</td>
<td>Reference Data Object</td>
<td>Method of Sharing</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>Receivables</td>
<td>Collectors</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Receivables</td>
<td>Lockbox</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Receivables</td>
<td>Memo Lines</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Receivables</td>
<td>Payment Terms</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Receivables</td>
<td>Remit To Address</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Receivables</td>
<td>Revenue Contingencies</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Receivables</td>
<td>Transaction Source</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Receivables</td>
<td>Transaction Type</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Advanced Collections</td>
<td>Collections Setups</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Advanced Collections</td>
<td>Dunning Plans</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Tax</td>
<td>Tax Classification Codes</td>
<td>Assignment to multiple sets, no common values allowed</td>
</tr>
<tr>
<td>Human Resources</td>
<td>Departments</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Human Resources</td>
<td>Jobs</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Human Resources</td>
<td>Locations</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Application Name</td>
<td>Reference Data Object</td>
<td>Method of Sharing</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>Human Resources</td>
<td>Grades</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Project Billing</td>
<td>Project and Contract Billing</td>
<td>Assignment to multiple sets, no common values allowed</td>
</tr>
<tr>
<td>Project Foundation</td>
<td>Project Accounting Definition</td>
<td>Assignment to one set only, no common values allowed</td>
</tr>
<tr>
<td>Project Foundation</td>
<td>Project Rates</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Order Management</td>
<td>Hold Codes</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Order Management</td>
<td>Orchestration Process</td>
<td>Assignment to one set only, with common values</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Application Name</th>
<th>Reference Data Object</th>
<th>Method of Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assets</td>
<td>Bonus Rules</td>
<td>Assignment to one set only, no common values allowed</td>
</tr>
<tr>
<td>Assets</td>
<td>Depreciation Ceilings</td>
<td>Assignment to one set only, no common values allowed</td>
</tr>
<tr>
<td>Assets</td>
<td>Depreciation Methods</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Assets</td>
<td>Asset Descriptions</td>
<td>Assignment to one set only, no common values allowed</td>
</tr>
<tr>
<td>Assets</td>
<td>Property Types</td>
<td>Assignment to one set only, with common values</td>
</tr>
<tr>
<td>Assets</td>
<td>Prorate Conventions</td>
<td>Assignment to one set only, no common values allowed</td>
</tr>
</tbody>
</table>
What reference data objects can be shared across cost organizations?
The following table contains the reference data objects for Oracle Fusion Cost Management that can be shared across cost organizations and the method in which the reference data for each is shared.

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

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

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

Define Enterprise: Manage Enterprise HCM Information
Define Enterprises

An enterprise is a collection of legal entities sharing 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. 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.

Enterprise Information for Non-HCM Users

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’re 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 aren't relevant outside of Oracle Fusion HCM.

Define Enterprise: Manage Locations

Locations

A location identifies physical addresses of a workforce structure, such as a department or a job. You create and manage locations using the Manage Locations task in the Workforce Structures work area under My Client Groups.

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 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.

Note the following:

- 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.
Oracle ERP Cloud
Implementing Common Features for Financials and Project Portfolio Management

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

Uploading Locations Using a Spreadsheet
If you have a list of locations already defined for your enterprise, you can upload them from a spreadsheet.

To use this option:

- Download a spreadsheet template
- Add your location information to the spreadsheet
- Upload directly to your enterprise configuration

You can upload the spreadsheet multiple times to accommodate revisions.

Related Topics
- Upload Workforce Structures Using a Spreadsheet

FAQs for Manage Locations

Why can't I see my location in the search results?
You can search for approved locations only. Also, if you created a location in Oracle Fusion Trading Community Model, then you can't access that location from Oracle Fusion Global Human Resources. For use in Oracle Fusion HCM, you must recreate the location from the Manage Locations page.
What happens if I select a geographic hierarchy node when I create or edit a location?
The calendar events that you created for the geographic node start to apply for the location and may impact the availability of worker assignments at that location. You manage locations using the Manage Locations task in the Workforce Structures work area.
The geographical hierarchy nodes available for selection on the Locations page display from a predefined geographic hierarchy.

Related Topics
- How an Employee's Schedule Is Determined

What happens if I select an inventory organization when I am 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 Oracle Fusion Global Human Resources, go to the Manage Locations page. Use the Manage Locations task in the Workforce Structures work area.
To appear on the Create or Edit Location pages, your inventory organization must be effective on today’s date and must exist in the location set that you selected.

Define Geographies

How Geography Structure, Hierarchy, and Validation 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, in the hierarchy of 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 format 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 to include 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

This topic describes geography structures and the tasks you can perform using geography structures.

A geography structure is a hierarchical grouping of geography types for a country. The following table describes the geography structure for the United States.

<table>
<thead>
<tr>
<th>Level</th>
<th>Geography Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>State</td>
</tr>
<tr>
<td>2</td>
<td>County</td>
</tr>
<tr>
<td>3</td>
<td>City</td>
</tr>
<tr>
<td>4</td>
<td>Postal Code</td>
</tr>
</tbody>
</table>

You can use the geography structure to relate geography types for a country and define geography types for a country.

Relate Geography Types for a Country

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 geography type Country is implicitly at the highest level of the geography structure with level as 1. The subsequent geography types that you add after country are numbered in sequence.

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. To quickly create country structure, you can copy a structure from another country and modify the geography types for the country.

Oracle ERP Cloud
Implementing Common Features for Financials and Project Portfolio Management

Chapter 4

Define Enterprise Structures

89
Define Geography Types for a Country

You can use any of the master reference geography types to create your geography structure. If required, you can create a geography type, before adding it to the country structure. Each geography type is added at a lower level to the current lowest level.

Note: You can’t delete geography types that have associated geography data. You can only delete the lowest level geography type of the country structure.

You can use a geography type that you create within the country structure for other country structures as well.

Geography Hierarchy

This topic describes geography hierarchy and various aspects of geography hierarchy.

Geography hierarchy is a data model that creates conceptual parent-child relationships between geographies. At the highest level of the geography hierarchy is country, which is the parent, and the hierarchy contains several child geographies. The following table shows sample parent-child relationships in a geography.

<table>
<thead>
<tr>
<th>California</th>
<th>Parent of San Mateo county</th>
</tr>
</thead>
<tbody>
<tr>
<td>San Mateo County</td>
<td>Parent of Redwood City</td>
</tr>
<tr>
<td>Redwood City</td>
<td>Parent of 94065</td>
</tr>
<tr>
<td>94065</td>
<td>Child</td>
</tr>
</tbody>
</table>

When you enter just 94065, the application determines that the postal code is in California and the corresponding city is Redwood City.

The application uses geography hierarchy information to facilitate business processes that rely on geography information, such as, tax calculation, order sourcing rules, and sales territory definition. The geography hierarchy information is centrally located and shared among other application offerings.

The geography hierarchy includes:

- Geography: Geography is a physical space with boundaries that’s a defined instance of a geography type, such as country, state, province or city. For example, San Jose is a geography of the City geography type.
- Geography type: Geography types are divisional grouping of user defined geographies, for example, Continent, Country Regions, and Tax Regions.
- Geography usage: Geography usage indicates how a geography type or geography is used in the application.
- Master reference geography hierarchy: The geography hierarchy data is considered the single source of reference for all geography related data such as geography types and geographies.

The geography usage for the entire hierarchy is the master reference, and defined geography types and geographies are the master reference geography types and geographies. For example, you can create geography types called State, City, and Postal Code. Then, you can rank the State as the highest level, City as the second level, and Postal Code as the lowest level within the country structure.
• User defined zones: User defined zones are a collection of geographical data, created from master reference data for a specific purpose. For example, while the territory zones are collections of master reference geographies ordered with a hierarchy, the tax and shipping zones are without a hierarchical grouping.

Geography Validation

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

Attribute Mapping

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. If the setup for master geography data is incomplete, then the geography data is either not imported or created. As a result, the list of values for the address attribute doesn't list any geography data.

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 isn't available in the geography hierarchy, you can't 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 must set up geography validation for those geography elements that you plan to use in your sales territories. Setting up validation also helps users fill in missing address information, and validate addresses during entry. For example, you can have users select states or other address elements from lists to ensure accuracy during entry, and you can have the application fill in missing values. For example, when the user enters a Postal Code, the application can retrieve the city and state.

You can specify whether a geography type will be included in geography validation. 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 are provided for the address element, but the address element isn’t be validated.

You need to verify that the default mapping between **Geography Type** and **Map to Attribute** is valid in the Geography Mapping and Validation region and update it if required when you define geography validation. Oracle recommends that you use the following valid mapping for the countries that GBG | Loqate supports:

<table>
<thead>
<tr>
<th>Country Name</th>
<th>Country Code</th>
<th>Geography Type</th>
<th>Map to Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andorra</td>
<td>AD</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Parroquia</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Settlement</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Angola</td>
<td>AO</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Provincia</td>
<td>• Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Municipio</td>
<td>• County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Comuna</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Localidad</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Argentina</td>
<td>AR</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Province</td>
<td>• Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Department</td>
<td>• County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Municipality</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Australia</td>
<td>AU</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• State</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• City</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Austria</td>
<td>AT</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Bundensland</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Bezirk</td>
<td>• County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Gemeinde</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Belgium</td>
<td>BE</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Gewest</td>
<td>• Additional address attribute 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Provincie</td>
<td>• Additional address attribute 2</td>
</tr>
<tr>
<td>Country Name</td>
<td>Country Code</td>
<td>Geography Type</td>
<td>Map to Attribute</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------</td>
<td>------------------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>Bolivia</td>
<td>BO</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Department</td>
<td>• Additional address attribute 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Provincia</td>
<td>• Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Canton</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Bosnia and Herzegovina</td>
<td>BA</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• District</td>
<td>• Additional address attribute 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Kanton</td>
<td>• Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Opcine</td>
<td>• County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Nasalje</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Brazil</td>
<td>BR</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• State</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• City</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Bulgaria</td>
<td>BG</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Oblast</td>
<td>• Additional address attribute 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Obshhina</td>
<td>• Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Settlement</td>
<td>• County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Postal code</td>
</tr>
<tr>
<td>Canada</td>
<td>CA</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Province</td>
<td>• Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• City</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Cayman Islands</td>
<td>KY</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Island</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• District</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Chile</td>
<td>CL</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Region</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Provincia</td>
<td>• County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Kommune</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>China</td>
<td>CN</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Province</td>
<td>• Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• City</td>
<td>• City</td>
</tr>
<tr>
<td>Country Name</td>
<td>Country Code</td>
<td>Geography Type</td>
<td>Map to Attribute</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>----------------------------------------------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>Croatia</td>
<td>HR</td>
<td>• Country</td>
<td>• Postal code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Zupanije</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Grad</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td></td>
</tr>
<tr>
<td>Cuba</td>
<td>CU</td>
<td>• Country</td>
<td>• Postal code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Provincia</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Municipio</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• City</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td></td>
</tr>
<tr>
<td>Cyprus</td>
<td>CY</td>
<td>• Country</td>
<td>• Postal code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Periochi</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Dimos</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td></td>
</tr>
<tr>
<td>Czech Republic</td>
<td>CZ</td>
<td>• Country</td>
<td>• Postal code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Kraj</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Okres</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Obec</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td></td>
</tr>
<tr>
<td>Denmark</td>
<td>DK</td>
<td>• Country</td>
<td>• Postal code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Amt</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Settlement</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td></td>
</tr>
<tr>
<td>Dominican Republic</td>
<td>DO</td>
<td>• Country</td>
<td>• Postal code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Provincia</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Municipio</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Town</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td></td>
</tr>
<tr>
<td>Ecuador</td>
<td>EC</td>
<td>• Country</td>
<td>• Postal code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Province</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Canton</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• City</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td></td>
</tr>
<tr>
<td>Estonia</td>
<td>EE</td>
<td>• Country</td>
<td>• Postal code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Maakond</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Vald</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td></td>
</tr>
<tr>
<td>Finland</td>
<td>FI</td>
<td>• Country</td>
<td>• Postal code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Laani</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Kunta</td>
<td></td>
</tr>
<tr>
<td>Country Name</td>
<td>Country Code</td>
<td>Geography Type</td>
<td>Map to Attribute</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>France</td>
<td>FR</td>
<td>- Country</td>
<td>- Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Region</td>
<td>- Additional address attribute 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Department</td>
<td>- State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Commune</td>
<td>- County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Postal Code</td>
<td>- City</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Postal code</td>
</tr>
<tr>
<td>Germany</td>
<td>DE</td>
<td>- Country</td>
<td>- Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bundesland</td>
<td>- State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Kreis</td>
<td>- County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Gemeinde</td>
<td>- City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Postal Code</td>
<td>- Postal code</td>
</tr>
<tr>
<td>Great Britain</td>
<td>GB</td>
<td>- Country</td>
<td>- Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- County</td>
<td>- State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Township</td>
<td>- County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Postal Code</td>
<td>- City</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Postal code</td>
</tr>
<tr>
<td>Greece</td>
<td>GR</td>
<td>- Country</td>
<td>- Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Periferia</td>
<td>- State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Nomi</td>
<td>- County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Dimotika</td>
<td>- City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Postal Code</td>
<td>- Postal code</td>
</tr>
<tr>
<td>Guadalupe</td>
<td>GP</td>
<td>- Country</td>
<td>- Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Commune</td>
<td>- State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Postal Code</td>
<td>- County</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- City</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Postal code</td>
</tr>
<tr>
<td>Guam</td>
<td>GU</td>
<td>- Country</td>
<td>- Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- State</td>
<td>- State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- County</td>
<td>- County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Municipality</td>
<td>- City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Village</td>
<td>- Postal code</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Additional address attribute 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Postal code</td>
</tr>
<tr>
<td>Guernsey</td>
<td>GG</td>
<td>- Country</td>
<td>- Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- City</td>
<td>- City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Parishes</td>
<td>- Additional address attribute 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Villages</td>
<td>- Additional address attribute 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Postal Code</td>
<td>- Postal Code</td>
</tr>
<tr>
<td>Hungary</td>
<td>HU</td>
<td>- Country</td>
<td>- Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Megye</td>
<td>- Additional address attribute 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Jaras</td>
<td>- Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Telepules</td>
<td>- County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Settlement</td>
<td>- City</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Postal Code</td>
</tr>
<tr>
<td>Country Name</td>
<td>Country Code</td>
<td>Geography Type</td>
<td>Map to Attribute</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>----------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>Iceland</td>
<td>IS</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Jaras</td>
<td>• Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Telepules</td>
<td>• County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Settlement</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>India</td>
<td>IN</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• State</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• City</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Indonesia</td>
<td>ID</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Province</td>
<td>• Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Regency</td>
<td>• County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• City</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Ireland</td>
<td>IE</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Province</td>
<td>• Additional address attribute 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Post County</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Post Town</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Localitat</td>
<td>• Additional address attribute 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Isles of Man</td>
<td>IM</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Localitat</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Settlement</td>
<td>• Additional address attribute 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Israel</td>
<td>IL</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• District</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Subdistrict</td>
<td>• County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• City</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Italy</td>
<td>IT</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Regione</td>
<td>• Additional address attribute 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Provincia</td>
<td>• Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Comune</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Jamaica</td>
<td>JM</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Parish</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Settlement</td>
<td>• City</td>
</tr>
<tr>
<td>Japan</td>
<td>JP</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td>Country Name</td>
<td>Country Code</td>
<td>Geography Type</td>
<td>Map to Attribute</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Prefecture</td>
<td>· State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· City</td>
<td>· City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Postal Code</td>
<td>· Postal code</td>
</tr>
<tr>
<td>Jordan</td>
<td>JO</td>
<td>· Country</td>
<td>· Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Muhafazat</td>
<td>· State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Liwa</td>
<td>· County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Tajma</td>
<td>· City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Postal Code</td>
<td>· Postal code</td>
</tr>
<tr>
<td>Kenya</td>
<td>KE</td>
<td>· Country</td>
<td>· Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· County</td>
<td>· State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Division</td>
<td>· City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Settlement</td>
<td>· Additional address attribute 2</td>
</tr>
<tr>
<td>Latvia</td>
<td>LV</td>
<td>· Country</td>
<td>· Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Rajons</td>
<td>· State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Piiseta</td>
<td>· City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Postal Code</td>
<td>· Postal code</td>
</tr>
<tr>
<td>Liechtenstein</td>
<td>LI</td>
<td>· Country</td>
<td>· Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Wahlkreis</td>
<td>· Additional address attribute 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Gemeinde</td>
<td>· State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· District</td>
<td>· City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Postal Code</td>
<td>· Postal code</td>
</tr>
<tr>
<td>Lithuania</td>
<td>LT</td>
<td>· Country</td>
<td>· Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· County</td>
<td>· State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Municipality</td>
<td>· County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Ward</td>
<td>· City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Postal Code</td>
<td>· Postal code</td>
</tr>
<tr>
<td>Luxembourg</td>
<td>LU</td>
<td>· Country</td>
<td>· Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· District</td>
<td>· Additional address attribute 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Canton</td>
<td>· State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Commune</td>
<td>· County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Settlement</td>
<td>· City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Postal Code</td>
<td>· Postal code</td>
</tr>
<tr>
<td>Malaysia</td>
<td>MY</td>
<td>· Country</td>
<td>· Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· State</td>
<td>· State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· District</td>
<td>· County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Subdistrict</td>
<td>· City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Postal Code</td>
<td>· Postal code</td>
</tr>
<tr>
<td>Malta</td>
<td>MT</td>
<td>· Country</td>
<td>· Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Region</td>
<td>· State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· Local Council</td>
<td>· City</td>
</tr>
<tr>
<td>Country Name</td>
<td>Country Code</td>
<td>Geography Type</td>
<td>Map to Attribute</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>------------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>Martinique</td>
<td>MQ</td>
<td>- Country</td>
<td>- Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Region</td>
<td>- State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Department</td>
<td>- County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Commune</td>
<td>- City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Postal Code</td>
<td>- Postal code</td>
</tr>
<tr>
<td>Mexico</td>
<td>MX</td>
<td>- Country</td>
<td>- Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- State</td>
<td>- State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Municipality</td>
<td>- County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Locality</td>
<td>- City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Postal Code</td>
<td>- Postal code</td>
</tr>
<tr>
<td>Netherlands</td>
<td>NL</td>
<td>- Country</td>
<td>- Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Province</td>
<td>- Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Gemeente</td>
<td>- County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Settlement</td>
<td>- City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Postal Code</td>
<td>- Postal code</td>
</tr>
<tr>
<td>New Zealand</td>
<td>NZ</td>
<td>- Country</td>
<td>- Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Region</td>
<td>- State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Town</td>
<td>- County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Postal Code</td>
<td>- City</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Postal code</td>
</tr>
<tr>
<td>Norway</td>
<td>NO</td>
<td>- Country</td>
<td>- Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Fylke</td>
<td>- Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Kommune</td>
<td>- County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Postal Code</td>
<td>- City</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Postal code</td>
</tr>
<tr>
<td>Oman</td>
<td>OM</td>
<td>- Country</td>
<td>- Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Muhafazah</td>
<td>- State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- District</td>
<td>- County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- City</td>
<td>- City</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Postal code</td>
</tr>
<tr>
<td>Peru</td>
<td>PE</td>
<td>- Country</td>
<td>- Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Region</td>
<td>- Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Provincia</td>
<td>- County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Distrito</td>
<td>- City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Postal Code</td>
<td>- Postal code</td>
</tr>
<tr>
<td>Poland</td>
<td>PL</td>
<td>- Country</td>
<td>- Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Wojewodztwo</td>
<td>- Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Powiat</td>
<td>- County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Gmina</td>
<td>- City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Postal Code</td>
<td>- Postal code</td>
</tr>
<tr>
<td>Portugal</td>
<td>PT</td>
<td>- Country</td>
<td>- Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Distrito</td>
<td>- Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Concelho</td>
<td>- County</td>
</tr>
<tr>
<td>Country Name</td>
<td>Country Code</td>
<td>Geography Type</td>
<td>Map to Attribute</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>-------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>Puerto Rico</td>
<td>PR</td>
<td>• Settlement</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Country</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Commonwealth</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Municipio</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Barrio</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td></td>
</tr>
<tr>
<td>Qatar</td>
<td>QA</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Municipality</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Zone</td>
<td>• County</td>
</tr>
<tr>
<td>Reunion Island</td>
<td>RE</td>
<td>• Country</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Region</td>
<td>• County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Department</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Commune</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td></td>
</tr>
<tr>
<td>Romania</td>
<td>RO</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Regiune</td>
<td>• Additional address attribute 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Judet</td>
<td>• Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Settlement</td>
<td>• County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Comuna</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Russia</td>
<td>RU</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Federal_District</td>
<td>• Additional address attribute 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Respublik</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Kray</td>
<td>• County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Oblast</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>San Marino</td>
<td>SM</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Comune</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Singapore</td>
<td>SG</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Region</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• District</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Slovakia</td>
<td>SK</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Kraj</td>
<td>• Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Okres</td>
<td>• County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Obec</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Slovenia</td>
<td>SI</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td>Country Name</td>
<td>Country Code</td>
<td>Geography Type</td>
<td>Map to Attribute</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>----------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>South Africa</td>
<td>ZA</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td>South Korea</td>
<td>KR</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td>Spain</td>
<td>ES</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td>Sri Lanka</td>
<td>LK</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td>Swaziland</td>
<td>SZ</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td>Sweden</td>
<td>SE</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td>Switzerland</td>
<td>CH</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td>Taiwan</td>
<td>TW</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td>Country Name</td>
<td>Country Code</td>
<td>Geography Type</td>
<td>Map to Attribute</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>----------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Thailand</td>
<td>TH</td>
<td>• Town</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Region</td>
<td>• Additional address attribute 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Changwat</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Amphoe</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Tunisia</td>
<td>TN</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Governorate</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Delegation</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal Code</td>
</tr>
<tr>
<td>Turkey</td>
<td>TR</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Sehir</td>
<td>• Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Ilce</td>
<td>• County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Kasaba</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>United Arab Emirates</td>
<td>AE</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Emirate</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• City</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Area</td>
<td>• Additional address attribute 2</td>
</tr>
<tr>
<td>United States</td>
<td>US</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• State</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• County</td>
<td>• County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• City</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Uruguay</td>
<td>UY</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Departamento</td>
<td>• State</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• City</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Vatican City</td>
<td>VA</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Municipal</td>
<td>• Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Settlement</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
<tr>
<td>Vietnam</td>
<td>VN</td>
<td>• Country</td>
<td>• Country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Region</td>
<td>• Additional address attribute 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tihn</td>
<td>• Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Quan</td>
<td>• County</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Thank Pho</td>
<td>• City</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Postal Code</td>
<td>• Postal code</td>
</tr>
</tbody>
</table>
Note: For either the tax or geography validation, don't skip more than one consecutive level unless you're 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 didn’t pass validation during address entry. For example, if the validation level is Error, then an address can’t be saved if the values don’t 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’s 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.

The Geography Dimension value in territories is derived from sell-to addresses of sales accounts. To use geography dimensions in territories, you must validate the geography elements in the addresses, such as state, city, and postal code. You can validate the address by enabling geography validation for each country using the Manage Geographies task. Perform the following in the Manage Geographies task:
- Enable at least one level in the geography hierarchy for geography validation.
- Enable geography validation for all geography levels that you intend to use for territory definition for each country.
- If needed, 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 set geography validation control to Error in the Manage Geography Validation page. This ensures that users can only use valid geography elements in addresses.

Note: If you have already created addresses before setting up geography validation for a country, you must enable geography validation and then execute the Run Maintain Geography Name Referencing task for that country. This validates all your geography elements.

Manage Geography Structures, Hierarchies, and Validation
This example shows how you can configure the geography structure, hierarchy, and validation for a country geography, using the United Kingdom country geography as an illustration.

The following table summarizes the key decisions for this scenario.

<table>
<thead>
<tr>
<th>Decisions to Consider</th>
<th>In This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy an existing country structure?</td>
<td>No, create a new country structure.</td>
</tr>
</tbody>
</table>
### Decisions to Consider

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

### Define the Geography Structure

You can 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 Geographies 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**.

### Define the Geography Hierarchy

You can add the geographies for the County and Post Town geography types using the geography hierarchy user interfaces to create the geography hierarchy for United Kingdom. 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. In the Geography Hierarchy section, click United Kingdom to highlight the table row, and click **Create**.
4. In the **Create County** page, Primary and Alternate Names section, enter Berkshire in the **Name** field.
5. Click **Save and Close**.
6. In the Geography Hierarchy section, click Berkshire to highlight the table row, and click **Create**.
7. In the **Create Post Town** page, Primary and Alternate Names section, enter Reading in the **Name** field.
8. Click Save and Close.
Define the Geography Validations

You can define the geography mapping and validation for the United Kingdom default address style format to specify the geography validations for the geography types you added to United Kingdom. You can 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. In the 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. Select the **Enable List of Values** and **Geography Validation** options.
5. For the Post Town geography type, click the **City** list item in the **Map to Attribute** field.
6. Select the **Geography Validation** option.
7. In the Geography Validation Control section, select **Error** in the **Geography Validation Level for Country** list.
8. Click **Save and Close**.

Geocoding

This topic explains geocoding and how to enable this option in the application.

Geocoding is the process of finding latitude and longitude coordinates from geographic data such as street addresses or postal codes. Once these coordinates are available, you can use the spatial services feature to identify points of interest, such as customer and contact addresses, in the vicinity. The application integrates the Geocoding feature with eLocation (http://elocation.oracle.com/maps_oracle_dot_com_main.html), which is a Geocoding service provided by Oracle.

By default, the Geocoding option is turned off in the application. You can enable the Geocoding option by going to the Setup and Maintenance work area and performing the following:

- Offering: Customer Data Management
- Functional Area: Enterprise Profile
- Task: Manage Geographies

If the Geocoding feature is enabled, the feature can be scheduled to run at regular time intervals. This ensures that newly created or updated locations are picked up and geocoded whenever you create or update an address using the user interface, web services, bulk import, or file-based import.

**Related Topics**

- What are Spatial Services

Set Up Geocoding

This procedure lists the steps to set up geocoding in Oracle Cloud applications.

Geocoding is a process that determines the latitude and longitude coordinates for a location. By default, geocoding is turned off in the application. You can use geocoding to display customers in the vicinity of a mobile address.
Enable Geocoding for a Country

To enable geocoding for a country, complete these steps:

1. In the Setup and Maintenance work area, go to the following:
   - Offering: Customer Data Management
   - Functional Area: Enterprise Profile
   - Task: Manage Geographies

2. On the Manage Geographies page, search for the country for which you want to enable geocoding. You can either search by the country name or country code.

3. Click Search. The search results for the matching country names are displayed.

4. Select the country for which you want to enable the geocoding option.

5. Select Geocoding Defined for the country.

Populate Location Latitude and Longitude Information

Once geocoding is enabled, you can schedule this feature to run at regular time intervals so that newly created or updated locations are picked up and geocoded. To schedule the geocoding feature to run at regular intervals, complete these steps:

1. Navigate to Tools > Scheduled Processes work area.


3. Click the Name list and search for Populate Location Latitude and Longitude Information, and then click OK.

4. Enter the parameters such as Start Date and End Date, and click Submit.

How You Import Geographies

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.

Note: You must use File-Based data Import in a cloud implementation because you can't populate the interface tables directly in a cloud implementation.

Consider the following when importing geographies:

- Oracle-licensed geography reference data
- File-based import option
- Geography loader process option
- Import object entity, interface table, and destination tables

Oracle-Licensed Geography Reference Data

Oracle Applications Cloud includes third-party master geography data for multiple countries that can be imported. You can import Oracle-licensed data from GBG | Loqate, for those countries where the data is available, such as the U.S. You can import geography data using the Manage Geographies task. Search for the country, and select Import.
**Geography Data** from the **Actions** menu. If the licensed data isn't available or already set up for a particular country, then the **Import Geography Data** action is disabled.

### 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.

To access the File-Based Data Import functionality, in the Setup and Maintenance work area, go to the following:

- **Offering:** Customer Data Management
- **Functional Area:** Enterprise Profile
- **Task:** Manage Import Activities

For more information, see the Importing Geographies chapter in the Oracle CX Understanding File-Based Data Import and Export for CX Sales and Service guide.

### Geography Loader Process Option
Populate the interface table with your import data, then to schedule the import of data from the interface table to the destination table, in the Setup and Maintenance work area, use the following:

- **Offering:** Customer Data Management
- **Functional Area:** Enterprise Profile
- **Task:** Run Geography Loader

### 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're 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, then, in the Setup and Maintenance work area, use the following to export the information:

- **Offering:** Customer Data Management
- **Functional Area:** Trading Community Foundation
- **Task:** Schedule Export Processes


The following table lists the object entity, the interface table, the destination tables, and the resulting application object.

<table>
<thead>
<tr>
<th>File-Based Import Entities</th>
<th>Interface Tables</th>
<th>Destination Tables</th>
<th>Application Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImpGeography</td>
<td>HZ_IMP_GEOGRAPHIES_T</td>
<td>HZ_GEOGRAPHIES</td>
<td>Geography</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HZ_GEOGRAPHY_IDENTIFIERS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HZ_GEOGRAPHY_TYPES_B</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HZ_HIERARCHY_NODES</td>
<td></td>
</tr>
</tbody>
</table>
Related Topics
- Overview of Implementing Customer Data Management
- Import Your Geography Data
- Import Your Territory Geographies Data

Create Countries

This procedure lists the steps to create countries in the application.

The countries are seeded in the application. If you're unable to find a specific country in the Manage Geographies page, then you can add it to the application.

**Note:** The application provides support for GBG | Loqate geography data for countries. For countries where GBG | Loqate geography data isn't available, you can purchase the geography data from a third-party data provider and load it into the application using File-Based Data Import. For more information, see the Importing Geographies chapter in the Oracle CX Understanding File-Based Data Import and Export for CX Sales and Service guide. If countries aren't available in the application, then use the procedure outlined in this topic to create them.

Perform the following steps to create a new country.

1. In the Setup and Maintenance work area, go to the following:
   - Offering: Customer Data Management
   - Functional Area: Application Extensions
   - Task: Manage Territories
2. On the Manage Territories page, click the **New** icon.
3. Enter the following details:
   - Territory Code: Enter a unique code for the territory.
   - Territory Name: Enter a unique name for the territory.
   - Description: Enter a description for the territory.
4. Click **Save and Close**.

   **Note:** After you have added a new country in the application, if you want to import the geography data for that country, then you must perform Step 5 to 10.

5. In the Setup and Maintenance work area, go to the following:
   - Offering: Customer Data Management
   - Functional Area: Enterprise Profile
   - Task: Manage Geographies
6. On the Manage Geographies page, click **Actions > Create Country**.
7. In the Create Country dialog box, select the name of the country and click **Save**.
8. Click **Done**.

   **Note:** You can now import the geography data for the newly created country using GBG | Loqate geography data available in the application. In case geography data for your country isn't available from GBG | Loqate, you can purchase the geography data from a third-party data provider and import it using File-Based Import.
Overview of Geography Import

A geography is any region with a boundary around it. 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 company-specific zones and addresses.

You can import geography data from an external data source into CX Sales and B2B Service using the File-Based Data Import feature.

Consider the following questions when importing your data:

- How does your legacy or source system store and represent the geography information when compared to CX Sales and B2B Service?
- Do you have to configure values in CX Sales and B2B Service to map your existing data to the Geography import object?
- Do you have to extend CX Sales and B2B Service to add attributes?
- What import features are available to import your business data?
- How do I verify my imported data?

Note: CX Sales and B2B Service ships with third-party master geography data for multiple countries. You can import geography data using the Manage Geographies task. Search for the country, and select Import Geography Data from the Actions menu. If the licensed data isn't available for a country, then the Import Geography Data action is disabled. For more information, see the procedure Replacing Existing Master Geography Data with Revised Oracle-Licensed Geography Data.

How Business Objects are Structured

You must understand how your geography data corresponds with the data in CX Sales and B2B Service to map your legacy data to the data in CX Sales and B2B Service.

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're importing the country structure of Australia, the country structure might be:

1. Country
2. State
3. County
4. Town
5. ZIP

Every import object is a collection of attributes that help to map your data to the Oracle Applications Cloud data and to support one-to-many relationships between the structural components of a geography.

You must know the attribute details of the import objects so that you can prepare your import data. You can use the import reference guide (File-Based Data Import for CXSales and B2B Service) files that contain:

- Attribute descriptions
- Values that populate attributes by default when you don’t provide values
• Validation information for each attribute

The validation information includes the navigation path to the task where you can define values in Oracle Application Cloud. For example, if you have values in your data that correlate to a choice list in Oracle Application Cloud, then the validation information provides the task name where you can define your values. For additional information, such as a list of reference guide file names and locations, see the topic How Geography Import Objects Work Together.

Configurable Attributes
Here is how you can configure the objects to import your legacy or source data:

• Use the Application Composer to design your object model extensions and to generate the required artifacts to register your extensions.
• Make the artifacts available for importing the object.

You can map these configurable attributes to your source file data. You can use the same source file to import both the configurable attributes and the standard import object attributes.

How You Import Geographies Using File-Based Data Import
You must first prepare a source data file to import geographies. The source file can be either an XML file or a text file, such as a CSV files.

You use the file import process to:
1. Read the data in your source file.
2. Populate the interface tables according to your mapping.
3. Import the data into the application destination tables.

The Define File Import Setup and Maintenance task list includes the tasks required to:

• Configure the import objects
• Create source-file mappings
• Schedule the import activities

You can also access these tasks from the Data Import and Export functional area of the Sales offering.

You submit file import activities for each import object. When you're creating a new geography, you use the Geography object to import your data. You must have 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.

How You Verify Your Imported Data
You can use the File Import activity reports to verify imported data. Alternatively, you can also navigate to the Geographies work area to view the geography information that you have imported.

Related Topics
• How File-Based Data Import Works
• About File-Based Import Documentation
How Geography Import Objects Work Together

You use the Geography import object to import geographies and objects related to the geography. This topic describes the Geography import object and introduces:

- Target import objects
- Target import object attributes
- Target import object attribute reference guide files used for evaluating and mapping source file data

Overview of Geography Target Import Objects

Use the Geography import object to import geographies and objects related to geographies. The Geography import object is split into smaller objects to organize the individual attributes of the geography.

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.

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.

Geography Target Import Object Attributes

Compare attributes that you want to import with the available target object attributes and their valid values. Use a reference file to evaluate your source data and Sales and B2B Service attributes for mapping and validation. See the File-Based Data Import for CX Sales and B2B Service guide (https://docs.oracle.com/en/cloud/saas/sales/20a/oefbs/index.html). See the topic for your import object, which includes links to reference files for target import objects. A reference guide file includes:

- Attribute descriptions
- Default values
- Validations for the attributes

Review the validation for each attribute to know if you need to do any setup tasks.

Define and Manage Import Mappings

You must define a mapping between your source data and a combination of the target object and target object attributes to import your source file data. To define and manage import mappings, do one of these:

1. Navigate to the following in the Setup and Maintenance work area:
   - Offering: Sales
   - Functional Area: Data Import and Export
   - Task: Manage File Import Mappings
2. Define the mapping when you define the import activity. Navigate to the following in the Setup and Maintenance work area:
   - Offering: Sales
   - Functional Area: Data Import and Export
Task: Manage File Import Activities

**Note:** If any of your source attributes doesn't have a corresponding target object attribute, then you can extend the Geography object. Review the Application Composer Extensibility features for the Geography object for more information.

### Reference Files for Geography Target Import Objects

To access reference files for this object's target import objects, see the File-Based Data Import for CX Sales and B2B Service guide ([https://docs.oracle.com/en/cloud/saas/sales/20a/oefbs/index.html](https://docs.oracle.com/en/cloud/saas/sales/20a/oefbs/index.html)). See the topic for your import object, which includes links to reference files for target import objects.

Here is a list of reference files and their target import objects.

<table>
<thead>
<tr>
<th>Target Import Object</th>
<th>Description</th>
<th>Attribute Reference File Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImpGeography</td>
<td>Contains information that saves a country's geography hierarchy details, such as geography type, geography code, and so on.</td>
<td>HZ_IMP_GEOGRAPHIES_T_Reference</td>
</tr>
</tbody>
</table>

**Related Topics**

- How File-Based Data Import Works
- About File-Based Import Documentation

### Example of Importing Geographies

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

The following table summarizes the key decisions that you must make in this scenario.

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

Summary of the Tasks
You perform the following steps to create an import activity and activate the import:

1. Determining what information is in the source file.
2. Creating and scheduling the import activity.
3. Monitoring the import results.

Before You Start

1. Ensure that the combination of the Source ID and Parent Source ID values is unique for each row of data within a single import. However, your source data files don’t 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, then the changed IDs will not affect the import.
2. Ensure that all 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 must 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, then your geography import creates 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. The source geography data files must include a unique Source ID value for each row of data and Parent Source ID value for the parent of that row of data. The Source or Parent Source IDs should not be longer than 18 characters.
2. You can structure your geography source data, as shown in the following table.

<table>
<thead>
<tr>
<th>Geography Level</th>
<th>Name</th>
<th>Source ID</th>
<th>Parent Source ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Country)</td>
<td>US</td>
<td>1</td>
<td>NA</td>
</tr>
<tr>
<td>2 (State)</td>
<td>CA</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>3 (County)</td>
<td>Alameda</td>
<td>111</td>
<td>11</td>
</tr>
<tr>
<td>4 (City)</td>
<td>Pleasanton</td>
<td>1111</td>
<td>111</td>
</tr>
<tr>
<td>4 (City)</td>
<td>Dublin</td>
<td>1112</td>
<td>111</td>
</tr>
</tbody>
</table>
Create and Schedule the Import Activity

You can create an import activity, enter the import details, and schedule the import. An import activity includes selecting the source file or file location, mapping the source file to the database, and scheduling the import.

1. Navigate to the following in the Setup and Maintenance work area:
   - Offering: Sales
   - Functional Area: Data Import and Export
   - Task: Manage File Import Activities

2. In the Manage Import Activities page, click **Create**.
3. In the Create Import Activity: Map Fields page, map each field from your source file to the target object and attribute, as shown in the following table.

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

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

4. Click **Next**.
5. In the Create Import Activity: Map Fields page, map each field from your source file to the database object and attribute, as shown in the following table.

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

![Note:](image)

**Note:** If you can't map the fields from your source file to the relevant target object, then see the import object spreadsheets.

6. **Click Next.**

7. In the Create Import Activity: Create Schedule page, select **Immediate** in the Schedule field so that the import will start as soon as you activate it.

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

8. **Click Next.**

**Monitor the Import Results**

You can monitor the processing of the import activity and view the completion reports for both successful records and errors.

1. In the Create Import Activity: Review and Activate page, verify your import details in the Import Details, File Details, Import Options, and Schedule sections. Update the import details if required by navigating to the previous screens using the **Back** link.

2. Confirm your import details, and click **Activate** to submit the import.

   After the import activity has finished, the Status field value changes to Completed.

**Overview of Country Structure Import**

You can import country structure data from an external data source into Sales and B2B Service 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 United States has the geography type of State at the topmost level, followed by the County, the City, and the Postal Code.

You can use the country structure to store information about:

- The relationships between geographies within a country
- The types of geographies for a country

Consider the following questions when importing your data:

- How does your legacy or source system store and represent the country structure information when compared to Sales and B2B Service?

---

<table>
<thead>
<tr>
<th>Column Header</th>
<th>Example Value</th>
<th>Ignore</th>
<th>Object</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source ID</td>
<td>10265</td>
<td>No</td>
<td>Imp Geography</td>
<td>Source ID</td>
</tr>
<tr>
<td>Parent Source ID</td>
<td>1053</td>
<td>No</td>
<td>Imp Geography</td>
<td>Parent Source ID</td>
</tr>
</tbody>
</table>
• Do you have to configure values in Sales and B2B Service to map your existing data to the Country Structure import object?

• Do you have to extend Sales and B2B Service to add attributes?

• What import features are available to import your business data?

• How do I verify my imported data?

How Business Objects Are Structured

You must understand how your country structure data corresponds with the data in Sales and B2B Service to map your legacy data to the data in Sales and B2B Service.

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’re importing the country structure of Australia, you need to follow this country structure:

1. Country
2. State
3. County
4. Town
5. ZIP

Import Object for the Country Structure

The import object for country structures is GeoStructureLevel.

How Business Object Attributes are Structured

Every import object is a collection of attributes that help to map your data to the Oracle Applications Cloud data and to support one-to-many relationships between the structural components of a country structure.

You must know the attribute details of the import objects so that you can prepare your import data. You can use the import reference guide (File-Based Data Import for CX Sales and B2B Service) files that contain:

• Attribute descriptions
• Values that populate attributes by default when you don’t provide values
• Validation information for each attribute

The validation information includes the navigation path to the task where you can define values in Oracle Application Cloud. For example, if you have values in your data that correlate to a choice list in Oracle Application Cloud, then the validation information provides the task name where you can define your values. For additional information, such as a list of reference guide file names and locations, see the topic How Country Structure Import Objects Work Together.

Configurable Attributes

Here is how you can configure the objects to import your legacy or source data:

• Use the Application Composer to design your object model extensions and to generate the required artifacts to register your extensions.

• Make the artifacts available for importing the object.

You can map these configurable attributes to your source file data. You can use the same source file to import both the configurable attributes and the standard import object attributes.
How You Import Country Structures Using File-Based Data Import

You must first prepare a source data file to import accounts. The source file can be either an XML file or a text file, such as a CSV file.

You use the file import process to:

1. Read the data in your source file.
2. Populate the interface tables according to your mapping.
3. Import the data into the application destination tables.

The Define File Import Setup and Maintenance task list includes the tasks required to:

- Configure the import objects
- Create source-file mappings
- Schedule the import activities

You can also access these tasks from the Data Import and Export functional area of the Sales offering.

You must have the Master Data Management Administrator job role to access and submit the import activities for country structures.

How You Verify Your Imported Data

You can use the File Import activity reports to verify imported data. Alternatively, you can also navigate to the Country Structure work area to view the country structure information that you have imported.

Related Topics

- How File-Based Data Import Works
- About File-Based Import Documentation

How Country Structure Import Objects Work Together

You use the Country Structure import object to import country structures and objects related to the country structure. This topic describes the Country Structure import object and introduces:

- Target objects
- Target import object attributes
- Target import object attribute reference guide files used for evaluating and mapping source file data

Overview of Country Structure Target Import Objects

Use the Country Structure import object to import country structures and objects related to country structures. The Country Structure import object is split into smaller objects to organize the individual attributes of the country structure. The target import objects in the Country Structure import object are grouped into information about the country structure and information about the other objects.

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.
Country Structure Target Import Object Attributes

Compare attributes that you want to import with the available target object attributes and their valid values. Use a reference file to evaluate your source data and attributes for mapping and validation. See the File-Based Data Import for CX Sales and B2B Service guide (https://docs.oracle.com/en/cloud/saas/sales/20a/oefbs/index.html). See the topic for your import object, which includes links to reference files for target import objects. A reference guide file includes:

- Attribute descriptions
- Default values
- Validations for the attributes

Review the validation for each attribute to know if you need to do any setup tasks.

Define and Manage Import Mappings

You must define a mapping between your source data and a combination of the target object and target object attributes to import your source file data. To define and manage import mappings, do one of these:

1. Predefine the mappings in the Setup and Maintenance work area:
   - Offering: Sales
   - Functional Area: Data Import and Export
   - Task: Manage File Import Mappings

2. Define the mapping when creating an import activity in the Setup and Maintenance work area:
   - Offering: Sales
   - Functional Area: Data Import and Export
   - Task: Manage File Import Activities

Note: If any of your source attributes doesn’t have a corresponding target object attribute, then you can extend the Country Structure object. Review the Application Composer Extensibility features for the Country Structure object for more information.

Reference Files for Country Structure Target Import Objects

To access reference files for this object's target import objects, see the File-Based Data Import for CX Sales and B2B Service guide (https://docs.oracle.com/en/cloud/saas/sales/20a/oefbs/index.html). See the topic for your import object, which includes links to reference files for target import objects.

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

Here is a list of reference files and their target import objects.

<table>
<thead>
<tr>
<th>Target Import Object</th>
<th>Description</th>
<th>Reference Guide File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImpGeoStructureLevel</td>
<td>Information that specifies a country's geography structure.</td>
<td>HZ_IMP_GEO_STRUCTURE LEVELS_Reference</td>
</tr>
</tbody>
</table>
How You Set up Address Cleansing

Address cleansing validates, corrects, and standardizes address information that you enter in the application. Address cleansing, unlike geography validation, validates both the geography attributes and the address line attributes.

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

You can specify the real-time address cleansing level for each country by choosing either of these options:

- **None**: Specifies no real time address cleansing.
- **Optional**: Provides option to cleanse addresses.

Once you have enabled address cleansing for a country, a **Verify Address** icon appears at address entry points in the application. Click the icon to perform address cleansing and receive a corrected, standardized address. If the application doesn't find a matching address, then an alert message is displayed.

FAQs for Define Geographies

**When do I define address cleansing?**

When address data entered into the application must conform to a particular format, 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're 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 can't delete the level. For example, if a state geography has been created for the United States country geography structure, then the State level can't 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's at a lower level to any geography in a geography hierarchy?

Select the geography that you want to create a geography at lower level, and then click the **Create** icon. This lets you create a geography for a geography type that's one level lower to the geography type you selected. The structure of the country's geography types are defined in the Manage Geography Structure page.

Define Legal Jurisdictions and Authorities

Overview

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

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

Jurisdictions

Jurisdiction is a physical territory such as a group of countries, country, state, county, or parish where a particular piece of legislation applies. French Labor Law, Singapore Transactions Tax Law, and US Income Tax Laws are examples of particular legislation that apply to legal entities operating in different countries' jurisdictions. Judicial authority may be exercised within a jurisdiction.

Types of jurisdictions are:

- Identifying Jurisdiction
- Income Tax Jurisdiction
- Transaction Tax Jurisdiction

Identifying Jurisdiction

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

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

Income Tax Jurisdiction

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

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

Legal Authorities

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

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

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

Create Legal Jurisdictions, Addresses and Authorities

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

Legal Jurisdictions

Create a legal jurisdiction by following these steps:

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

Legal Addresses for Legal Entities and Reporting Units

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

1. Navigator > Setup and Maintenance > Manage Legal Address > Go to Task.
2. Select Create.
4. Enter Address Line 1, Oracle Parkway.
5. Optionally enter Address Line 2, and Address Line 3.
6. Enter or Select the postal code, 94065.
7. Select Geography 94065 and Parent Geography Redwood Shores, San Mateo, CA.
9. OK.
10. Save and Close.

Legal Authorities
Create a legal authority by following these steps:

1. Navigator > Setup and Maintenance > Manage Legal Authorities > Go to Task.
2. Enter the Name, California Franchise Tax Board.
3. Enter the Tax Authority Type, Reporting.
   
   Note: Create an address for the legal authority.
4. Select Create.
5. The Site Number is automatically assigned.
6. Optionally enter a Mail Stop.
7. Select Country, United States
8. Enter Address Line 1, 121 Spear Street, Suite 400.
9. Optionally enter Address Line 2, and Address Line 3.
10. Enter or Select the postal code, 94105.
12. OK.
14. Optionally click the One-Time Address check box.
15. The From Date displays today's date. Update if necessary.
16. Optionally enter a To Date to indicate the last day the address can be used.
   
   Note: You can optionally enter Address Purpose details.
17. Select Add Row.
18. Select Purpose.
19. The Purpose from Date will default to today's date.
20. Optionally enter a Purpose to Date.
21. OK.
22. Save and Close.

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

Legal Entity
Create a legal entity by following these steps:

1. Navigator > Setup and Maintenance > Manage Legal Entity > Go to Task.
2. Accept the default Country, United States.
3. Enter Name, InFusion USA West.
4. Enter Legal Entity Identifier, US0033.
5. Optionally enter Start Date. When the start date is blank the legal entity is effective from the creation date.
6. Optionally enter an End Date.
7. Optionally, if your legal entity should be registered to report payroll tax and social insurance, select the Payroll statutory unit check box.
8. Optionally, if your legal entity has employees, select the Legal employer check box.
9. Optionally, if this legal entity is not a payroll statutory unit, select an existing payroll statutory unit to report payroll tax and social instance on behalf of this legal entity.
10. Enter the Registration Information
11. Accept the default Identifying Jurisdiction, United States Income Tax.
12. Search for and select a Legal Address, 500 Oracle Parkway, Redwood Shores, CA 94065.
   The legal address must have been entered previously using the Manage Legal Address task.
13. OK.
14. Optionally enter a Place of Registration.
15. Enter the EIN or TIN.
16. Enter the Legal Reporting Unit Registration Number.
17. Save and Close.
18. Navigator > Setup and Maintenance > Define Legal Entries > Manage Legal Entity > Select to set scope.
19. Select the Manage Legal Entity.
20. In the *Legal Entity list, select Select and Add.
21. Click Apply and Go to Task.
22. Select your legal entity.
23. Save and Close.
   This sets the scope for your task list to the selected legal entity.
24. Save and Close.

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

1. Navigator > Setup and Maintenance > Manage Legal Entity Registrations: Verify that the Legal Entity scope value is set correctly.
2. Go to Task.
3. Select Create.
4. Enter Jurisdiction.
5. Enter Registered Address.
6. Enter Registered Name.
7. Optionally enter Alternate Name, Registration Number, Place of Registration, Issuing Legal Authority, and Issuing Legal Authority Address, Start Date, and End Date.
8. Save and Close.

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

1. Navigator > Setup and Maintenance > Define Legal Reporting Unit. > Manage Legal Reporting Unit. Verify that the Legal Entity scope value is set correctly.
2. Go to Task
3. Select Create.
4. Enter Territory, United States.
5. Enter Name.
6. Optionally enter a Start Date.
7. Enter Registration Information.
8. Search for and select Jurisdiction.
9. Enter Main Legal Reporting Unit information.
10. Select the value Yes or No for the Main Legal Reporting Unit. Set value to yes only if you are creating a new main (primary) legal reporting unit.
11. Enter the Main Effective Start Date, 1/1/11.
12. Save and Close.

Define Legal Entities: Manage Legal Entity

Overview

A legal entity is a recognized party with rights and responsibilities given by legislation. Legal entities have the following rights and responsibilities to:

- Own property
- Trade
- Repay debt
- 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
- Minimizing the enterprise's tax liability
- 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, 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
- 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), which enforces statutory reporting requirements for public corporations.

Individual entities privately held or held by public companies don't 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.

Model Legal Entities

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. These registers are part of the Oracle Fusion Trading Community Architecture.

When your legal entities are trading with each other, represent them as legal entities and 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.

Several decisions you should consider when you create legal entities.

- The importance of using legal entity on 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 Using Legal Entities on 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, a sales order creates an obligation on the legal entity that books the order and promises to deliver the goods on the acknowledged date. The creation also creates an obligation on the purchaser to receive and pay for those goods. Contract law in most countries contains statutes that state 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.
- 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's processing the transaction. For example, Business Unit ACM UK has a default legal entity of InFusion UK Ltd. When a purchase order is placed in ACM UK, the legal entity InFusion UK Ltd is legally obligated to the supplier. Oracle Fusion Procurement, Oracle Fusion Project Portfolio Management, 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 wanted, 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 one segment 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 with 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. This solution is used 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're disposing of a part of a legal entity and want to isolate the assets and liabilities for that entity.

Implementing multiple balancing segments requires every journal entry that isn't balanced by division or business unit, to generate balancing lines. You can't change to multiple balancing segments after you begin using the ledger because your historical data isn't balanced by the new balancing segments. Restating historical data must be done at that point.

If your enterprise regularly spins off businesses or holds managers accountable for utilization of assets, identify the business with a balancing segment value. If you account for each legal entity in a separate ledger, no requirement exists to identify the legal entity with a balancing segment value.

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 the portion isn't a separate legal entity. If you don't map legal entities sharing the same ledger to balancing segments, you can't distinguish them using intercompany functionality or track 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 feature to create intercompany entries automatically 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.

Tip: In the Oracle Fusion Supply Chain applications, you can model intercompany relationships using business units, from which legal entities are derived.

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

Define Legal Entities: Manage Legal Entity HCM Information

How Legal Employers Work with Payroll Statutory Units and Tax Reporting Units

You can designate legal entities as *legal employers* and *payroll statutory units*, which makes them available for use in Oracle Fusion Human Capital Management (HCM). You can have only one legal entity that's also a payroll statutory unit and legal employer, or multiple legal entities, payroll statutory units and legal employers.

Payroll statutory units and tax reporting units share a parent child relationship with the payroll statutory unit being a parent of a tax reporting unit.

Legal Employers and Payroll Statutory Units

You can designate payroll statutory units to group legal employers to do statutory calculations at a higher level. For example, you can use payroll statutory units for court orders, or to calculate the United Kingdom (UK) statutory sick pay. A legal employer can exist independently of an enterprise or be a part of a payroll statutory unit. There can be many legal employers belonging to a payroll statutory unit, but a legal employer can belong only to one payroll statutory unit.

Legal Employers and Tax Reporting Units

Tax reporting units are indirectly associated with a legal employer through the payroll statutory unit. One or more tax reporting units can be used by a single legal employer, and a tax reporting unit can be used by one or more legal employers.

For example, if a single tax reporting unit is linked to a payroll statutory unit and two legal employers are associated with this payroll statutory unit, then both legal employers are associated with the tax reporting unit. Use the Manage Legal Reporting Unit HCM Information task to designate an existing legal reporting unit as a tax reporting unit. You need to select a parent payroll statutory unit when you create a legal reporting unit belonging to a legal employer (that isn't a payroll statutory unit as well). Next, you need to designate the legal reporting unit as a tax reporting unit and select the legal employer.

Related Topics

- Examples of HCM Organization Models
- What's a tax reporting unit
FAQs for Manage Legal Entity HCM Information

What's a legal employer?
A legal employer is a legal entity that employs workers. You define a legal entity as a legal employer in the Oracle Fusion Legal Entity Configurator.

The legal employer is captured at the work relationship level, and all assignments within that relationship are automatically with that legal employer. Legal employer information for worker assignments is also used for reporting purposes.

What's a payroll statutory unit?
Payroll statutory units are legal entities that are responsible for paying workers, including the payment of payroll tax and social insurance. A payroll statutory unit can pay and report on payroll tax and social insurance on behalf of one or many legal entities, depending on the structure of your enterprise. For example, if you're a multinational, multiple company enterprise, then you register a payroll statutory unit in each country where you employ and pay people. You can optionally register a consolidated payroll statutory unit to pay and report on workers across multiple legal employers within the same country. You associate a legislative data group with a payroll statutory unit to provide the correct payroll information for workers.

Define Legal Entities: Manage Legal Entity Tax Profile

Party Tax Profiles
A tax profile is the body of information that relates to a party's transaction tax activities. A tax profile can include main and default information, tax registration, tax exemptions, party fiscal classifications, tax reporting codes, configuration options, and service subscriptions.

Set up tax profiles for the following parties involved in your transactions:

- First parties
- Third parties
- Tax authorities

First Parties
Set up tax profiles for your first-party legal entities, legal reporting units, and business units.

First-party legal entities identify your organization to the relevant legal authorities, for example, a national or international headquarters. Legal entities let you model your external relationships to legal authorities more accurately. The relationships between first-party legal entities and the relevant tax authorities normally control the setup of the transaction taxes required by your business. In most circumstances, the tax setup is used and maintained based on the configuration of the legal entity. Enter the default information, party fiscal classifications, tax reporting codes, and configuration options for your legal entities. You can also specify if you're using the tax services of an external service provider for tax calculation.

First-party legal reporting units identify each office, service center, warehouse, and any other location within the organization with a tax requirement. A legal reporting unit tax profile is automatically created for the headquarter legal
entity. Set up additional legal reporting unit tax profiles for those needed for tax purposes. For legal reporting units, enter the default information, tax registrations, party fiscal classifications, and tax reporting codes. Also, define tax reporting details for your VAT and global tax reporting needs for tax registrations of tax regimes that allow this setup.

Business units organize your company data according to your internal accounting, financial monitoring, and reporting requirements. To help you manage the tax needs of your business units, you can use the business unit tax profile in either of two ways:

- Indicate that business unit tax setup is used and maintained based on the configuration of the associated legal entity at transaction time. The tax setup of the associated legal entity setup is either specific to the legal entity or shared across legal entities using the Global Configuration Owner setup.
- Indicate that tax setup is used and maintained by a specific business unit. Create configuration options for the business unit to indicate that the subscribed tax content is used for the transactions created for the business unit.

For business units that maintain their own setup, enter the default information, tax reporting codes, configuration options, and service providers as required.

### Third Parties

Set up third-party tax profiles for parties with the usage of customer, supplier, and their sites. Enter the default information, tax registrations, party fiscal classifications, and reporting codes required for your third parties or third-party sites. You can set up tax exemptions for your customers and customer sites.

Banks are also considered third parties. When a bank is created, the tax registration number specified on the bank record is added to the party tax profile record in Oracle Fusion Tax. You can't modify the party tax profile for a bank as it’s view only. You can only modify the bank record.

![Note: You don’t need to set up party tax profiles for third parties. Taxes are still calculated on transactions for third parties that don’t have tax profiles.](image)

### Tax Authorities

Set up a tax authority party tax profile using the Legal Authorities setup task. The tax authority party tax profile identifies a tax authority party as a collecting authority or a reporting authority or both. A collecting tax authority manages the administration of tax remittances. A reporting tax authority receives and processes all company transaction tax reports.

The collecting and reporting tax authorities appear in the corresponding list of values on all applicable Oracle Fusion Tax pages. All tax authorities are available in the list of values as an issuing tax authority.

**Related Topics**

- Considerations for Specifying Third-Party Tax Profile Options
- When does a party tax profile get created for a third party

### Considerations for Specifying First-Party Tax Profile Options

Set up first-party tax profiles for all legal entities, legal reporting units, and business units in your organization that have a transaction tax requirements. How you set up your first parties can impact the tax calculation on your transactions.

The first-party tax profile consists of:

- Defaults and controls: Applicable to legal entities and legal reporting units. Business units that use their own tax setup don’t have defaults and controls.
• **Tax registrations**: Applicable to legal reporting units.
• **Party fiscal classifications**: Applicable to legal entities and legal reporting units.
• Tax reporting codes: Applicable to legal entities, legal reporting units, and business units who don’t use the tax setup of the legal entity.
• Configuration options: Applicable to legal entities and business units who don’t use the tax setup of the legal entity.
• Service subscriptions: Applicable to legal entities and business units who don’t use the tax setup of the legal entity.

### Defaults and Controls

The following table describes the defaults and controls available at the first-party tax profile level:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Set as self-assessment (reverse charge)</strong></td>
<td>Automatically self-assess taxes on purchases.</td>
</tr>
<tr>
<td><strong>Rounding Level</strong></td>
<td>Perform rounding operations on the:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Header</strong>: Applies rounding to calculated tax amounts once for each tax rate per invoice.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Line</strong>: Applies rounding to the calculated tax amount on each invoice line.</td>
</tr>
<tr>
<td><strong>Rounding Rule</strong></td>
<td>The rule that defines how the rounding must be performed on a value involved in a taxable transaction. For example, up to the next highest value, down to the next lowest value, or nearest.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: If you defined a rounding precedence hierarchy in the configuration owner tax option settings for the combination of configuration owner and event class, Oracle Fusion Tax considers the rounding details in the applicable tax profile.</td>
</tr>
<tr>
<td><strong>Set Invoice Values as Tax Inclusive</strong></td>
<td>This first party intends to send or receive invoices with invoice line amount inclusive of the tax amount.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: This option overrides the tax inclusive handling setting at the tax level, but not at the tax rate level.</td>
</tr>
</tbody>
</table>

### Tax Registrations

Set up a separate tax registration to represent each distinct registration requirement for a first-party legal reporting unit. Oracle Fusion Tax uses tax registrations in tax determination and tax reporting. If your first party has more than one tax registration in the same tax regime, then the application considers the tax registration in the order: tax jurisdiction; tax; tax regime.

You must enable the **Use tax reporting configuration** option on the first-party tax regime to allow entry of global tax reporting configuration details during tax registration setup for legal reporting units for these tax regimes.
Party Fiscal Classifications
If applicable, associate first-party fiscal classification codes with this party. The party fiscal classification codes you enter become part of tax determination for invoices associated with this party. Specify start and end dates to control when these fiscal classifications are applicable for this party and transaction.

For legal entities, you can view the associated legal classifications that were assigned to the tax regime defined for this first party. The legal classifications are used in the tax determination process, similar to the party fiscal classifications.

Tax Reporting Codes
Set up tax reporting types to capture additional tax information on transactions for your tax reports for your first parties. Depending on the tax reporting type code, you either enter or select a tax reporting code for this party. Specify start and end dates to control when these tax reporting codes are applicable.

Configuration Options
The legal entities and business units in your organization are each subject to specific sets of tax regulations as designated by the tax authorities where you do business. Use configuration options to associate legal entities and business units with their applicable tax regimes. You can set up tax configuration options when you create a tax regime or when you create a party tax profile. Both setup flows display and maintain the same party and tax regime definitions.

Service Subscriptions
You can use a service subscription to reference a specific transaction tax offering or offerings provided by an external tax partner. The transaction tax offering provided by an external tax partner can be related to content, calculation services, or both. Oracle Fusion Tax supports the use of transaction tax offerings provided by external tax partners for transaction tax calculation processing. Depending on the specific depth and scope of an individual tax partner’s offerings, you can use either Oracle Fusion Tax or a Partner Tax Application to perform the transaction tax calculation.

Related Topics
- Tax Registrations
- Tax Configuration Options
- Considerations for Setting Tax Reporting Configuration Controls for VAT
- Party Information

FAQs for Manage Legal Entity Tax Profile

When does a party tax profile get created for a legal entity?
A legal entity party tax profile is autocreations when you create a legal entity record.
You can also use other methods to create a legal entity party tax profile. For example, it’s created when a back-end process creates a legal entity, but only when you do these:

- Save a tax regime to which the legal tax entity subscribes.
- Save the configuration owner tax option that's defined for the legal entity.

You can also create one manually. Just use the Create Legal Entity Tax Profile page. You can also edit the autogenerated tax profile with relevant tax information.
Define Legal Entities: Define Legal Reporting Units

Plan Legal Reporting Units

Each of your legal entities has at least one legal reporting unit. Some legal reporting units can also be referred to as establishments. You can define either domestic or foreign establishments. Define legal reporting units by physical location, such as sales offices. For example, set up legal reporting units to represent your company and its offices for tax reporting.

Planning Legal Reporting Units

Plan and define your legal reporting units at both the local and national levels if you operate within the administrative boundaries of a jurisdiction that’s more granular than country. For example, your legal entity establishes operations in a country that requires reporting of employment and sales taxes locally as well as nationally. Therefore, you need more than one legally registered location to meet this legal entity’s reporting requirements in each area. Additionally, legal entities in Europe operate across national boundaries, and require you to set up legal reporting units for the purposes of local registration in each country. There can be multiple registrations associated with a legal reporting unit. However, only one identifying registration can be defined by the legal authority used for the legal entity or legal reporting unit and associated with the legal reporting unit.

Define Business Units: Manage Business Units

Business Units

A business unit is a unit of an enterprise that performs one or many business functions that can be rolled up in a management hierarchy. A business unit can process transactions on behalf of many legal entities. Normally, it has 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 do the following:

- Assign your business units to one primary ledger. For example, if a business unit is processing payables invoices, then it must post to a particular ledger. This assignment is required for your business units with business functions that produce financial transactions.
- Use a business unit as a securing mechanism for transactions. For example, if you run your export business separately from your domestic sales business, then 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 provides the following advantages:

- Enables flexible implementation
- Provides consistent entity that controls and reports on transactions
- Shares 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 share reference data, such as payment terms and transaction types, across business units, or you can have each business unit manage its own set depending on the level at which you want to enforce common policies.

In summary, use business units for:

- Management reporting
- Transaction processing
- Transactional data security
- Reference data sharing and definition

Brief Overview of Business Unit Security

A number of Oracle Fusion Applications use business units to implement data security. You assign roles like Accounts Payable Manager to users to permit them to perform specific functions, and you assign business units for each role to users to give them access to data in those business units. For example, users who have been assigned a Payables role for a particular business unit, can perform the function of payables invoicing on the data in that business unit. Roles can be assigned to users manually using the Security Console, or automatically using provisioning rules. Business Units can be assigned to users using the Manage Data Access for Users task found in Setup and Maintenance.

Define Business Units: Assign Business Unit Business Function

Business Functions

A business unit can perform many business functions in Oracle Fusion Applications.

Business Functions

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

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

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

**Note:** This hierarchy definition isn't required in the setup of your applications, but is a recommended best practice.

Your enterprise procedures can require a manager of a business unit to have responsibility for their profit and loss statement. In such cases, any segment that allows the identification of associated revenue and costs can be used as a profit center identification. The segment can be qualified as the Cost Center Segment.

However, there are cases where a business unit is performing only general and administrative functions, in which case your manager's financial goals are limited to cost containment or recovering of service costs. For example, if a shared service center at the corporate office provides services for more commercially-oriented business units, it doesn't show a profit and therefore, only tracks its costs.

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

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

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

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

**Business Unit Hierarchy: Example**

For example, your InFusion America Company provides:

• Air quality monitoring systems through your division InFusion Air Systems
Oracle ERP Cloud
Implementing Common Features for Financials and Project Portfolio Management

Chapter 4
Define Enterprise Structures

- Customer financing through your division InFusion Financial Services

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

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

The following figure shows an example of a business unit hierarchy.

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

Define Business Units: Manage Service Provider Relationships
Shared Service Centers

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

Service Provider Model

The service provider model centralizes the following business functions:

- **Procurement**
  - Services business units that enable the Requisitioning business function.
  - Processes requisitions and negotiates supplier terms for client business units.

- **Payables Payment**
  - Services business units that enable the Payables Invoicing business function.
  - Processes payments for client business units.

- **Customer Payments**
  - Services business units that enable the Billing and Revenue Management business function.
  - Processes payments for the transactions of client business units assigned the Billing and Revenue Management business function.

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

Guidelines for Shared Service Centers

Oracle Fusion Applications supports shared service centers in two ways. First, with business unit security, which allows your shared service centers personnel to process transactions for other business units called clients.

Second, the service provider model expands on this capability to allow a business unit and its personnel in a shared service center to work on transactions of the client business units. It is possible to view the clients of a service provider business unit, and to view service providers of a client business unit.

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

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

Security

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

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

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

Benefits

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

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

Service Provider Models

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

Procurement Example

The Oracle Fusion Procurement product family has taken advantage of the service provide model by defining outsourcing of the procurement business function. Define your business units with requisitioning and payables invoicing business functions as clients of your business unit with the procurement business function. Your business unit responsible for the procurement business function takes care of supplier negotiations, supplier site maintenance, and purchase order processing on behalf of your client business units. Subscribe your client business units to the supplier sites maintained by the service providers, using a new procurement feature for supplier site assignment.
In the InFusion example, business unit four (BU4) serves as a service provider to the other three business units (BU1, BU2, and BU3.) BU4 provides the corporate administration, procurement, and human resources (HR) business functions, thus providing cost savings and other benefits to the entire InFusion enterprise.

Define Business Units: Specify Contract Management Business Function Properties

Customer Contracts Business Unit Setup

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

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

- Enable related accounts
- Set currency conversion details

Note: You must select a default currency in the customer or supplier business function properties page, if not populated automatically from the ledger assigned to the business unit in the assign business function setup task.
• Manage project billing options
• Set up clause numbering
• Set up the Contract Terms Library
  The setup options available for the Contract Terms Library are applicable to both customer and supplier contracts, and are described in the business unit setup topic for the Contract Terms Library. That topic is available as a related link to this topic.

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

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

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

In the Invoice Currency to Ledger Currency region:
• Enter invoice transaction conversion details if the invoice and ledger currencies can be different.
• Enter revenue transaction conversion details if the revenue and ledger currencies can be different for as-incurred and rate-based revenue.

Managing Project Billing Options
The options available for selection in the Project Billing region control the behavior of project invoicing and revenue recognition for contracts with project-based work. Project billing can act differently for external contracts (customer billing) or intercompany and interproject contracts (internal billing).

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

There are two sets of the following options, one for customer billing and a second for internal billing:
• Select an invoice numbering method, either Manual or Automatic. The invoice numbering method is the method that Oracle Fusion Receivables uses to number its invoices, upon release of draft invoices from Project Billing.
  o If the invoice numbering method is Manual, then select an invoice number type, which sets the type of Receivables invoice numbers that are allowed. Valid values are Alphanumeric and Numeric.
  o If the invoice numbering method is Automatic, then enter the next invoice number to use when generating Receivables invoice numbers.
• Select the Receivables batch source to use when transferring invoices to Receivables.

Set this option only for customer billing:

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

Managing Clause Numbering
You can choose to number clauses manually or automatically.

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

Supplier Contracts Business Unit Setup
You can specify a variety of business function settings for supplier contracts in a specific business unit using the Specify Supplier Contract Management Business Function Properties task, available by selecting Setup and Maintenance from the Navigator and searching on the task name.

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

**Note:** The customer must select a default currency in the customer or supplier business function properties page, if not automatically populated from the ledger assigned to the business unit in the assign business function setup task.

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

Set Up Supplier Ship-to Organization, Ship-to Location and Bill-to Location
To set up ship-to organization, create an Inventory Organization by using these steps:

1. Navigate to Setup and Maintenance.
2. Click Search in the Tasks side panel and type Manage Inventory Organizations. Then, select it from the options listed.
3. Click Create in the results table.
4. Enter Name and Organization, and select Management Business Unit and Legal Entity information from the Drop Down list. Then, click Next.
5. In the General tab, select Schedule and Item Master Organization information. Click Save and Close.

To set up a ship-to and bill-to location, create an Inventory Organization Location by using the following steps:

1. Search for Manage Inventory Organization Locations in the Tasks side panel.
2. Click Manage Inventory Organization Locations on the Search page.
3. Click Create in the results table.
4. Enter mandatory fields and select Inventory Organization in the drop-down list.
5. Click Submit.
Contract Terms Library Business Unit Setup

You can specify a wide variety of Contract Terms Library settings for either customer or supplier contracts within each business unit, by using either the Specify Customer Contract Management Business Function Properties or the Specify Supplier Contract Management Business Function Properties tasks. These tasks are available in the Setup and Maintenance work area as part of the Enterprise Contracts offering in the Enterprise Contracts Base functional area.

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

- Enable clause and template adoption.
- Set the clause numbering method.
- Set the clause numbering level for automatic clause numbering of contracts.
- For a contract with no assigned ledger or legal entity, set the document sequence to Global or Business Unit level.
- Enable the Contract Expert enabling feature.
- Specify the layout for printed clauses and contract deviation reports.

Enabling Clause Adoption

Do this to use clause adoption in your implementation:

1. Specify a global business unit
   
   Designate one of the business units in your organization as the global business unit by selecting the Global Business Unit option. This makes it possible for the other local business units to adopt and use approved content from that global business unit. If the Global Business Unit option isn't available for the business unit you're setting up, it means you have already designated a business unit as global.

2. Enable automatic adoption
   
   If you’re implementing the adoption feature, you can have all the global clauses in the global business unit automatically approved and available for use in the local business by selecting the Autoadopt Global Clauses option. If you don’t select this option, the employee designated as the Contract Terms Library Administrator has to approve all the global clauses, before they can be adopted and used in the local business unit. This option is available only for local business units.

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

4. Adopt global clauses for new business unit
   
   If you’re creating a new local business unit and have to adopt existing global clauses, run the Adopt Global Clauses for a New Business Unit process. Refer to the Enterprise Scheduler processes topic for more information.

Setting Clause Numbering Options

You can set up automatic clause numbering for the clauses in the business unit by selecting Automatic in the Clause Numbering field and setting the clause numbering level. Then select the appropriate clause sequence category for
the specified numbering level. You must have previously set up document sequences for the document sequence categories of global, ledger, and business unit. If clause numbering is manual, contract terms library administrators must enter unique clause numbers each time they create a clause.

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

### Set Up Consistent Clause Numbering

When you skip numbering and suppress the title for a clause or section, the numbering format may look different when you download the contract. You get the difference in numbering format because your downloaded contract has the numbering format of Microsoft Word. To enable Microsoft Word numbering format for your contract in the Enterprise Contracts UI, you must set the **Consistent Microsoft Word Numbering for Contract Terms Enabled** profile option as **Yes**.

If this profile option is set as **No**, you may see inconsistent numbering when you download the contract as the Microsoft Word document.

Here are the steps to enable the profile option:

1. In the Setup and Maintenance work area, go to the **Manage Administrator Profile Values** task.
2. On the Manage Administrator Profile Values page, search for and select the **Consistent Microsoft Numbering for Contract Terms Enabled** profile option.
3. Set the profile option to **Yes**.
4. **Save and Close**.

### Enable Contract Expert

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

### Specify Printed Clause and Deviations Report Layouts

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

- The printed contract terms
  
  Enter the RTF file you want used for formatting the printed clauses in the **Clause Layout Template** field.

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

### Related Topics

- How Business Unit Affects Clauses and Other Objects in the Library
- How Contract Expert Works
- Contract Printing and Layout Templates
5 Define Security

Overview of Security

Oracle Enterprise Resource Planning Cloud (Oracle ERP Cloud) and Oracle Supply Chain Management Cloud (Oracle SCM Cloud) are secure as delivered; they limit access to one initial setup user. To enable application users to access application functions and data, you perform tasks in these task lists, as appropriate:

- Define Security for Financials
- Define Security for Procurement
- Define Users and Security for Product Management
- Define Security for Project Execution Management
- Manage Data Access for Users
- Define Security for Supply Chain Management

This topic introduces the tasks in these task lists. For more information on ERP and SCM security setup and task instructions, see these guides:


Note: You can perform most tasks in these task lists both during implementation, and later as requirements emerge.

Manage Job Roles

The Oracle ERP Cloud and Oracle SCM Cloud security reference implementations provide many predefined job roles. You can perform the Manage Job Roles task to:

- Review the role hierarchy and other properties of a job or abstract role.
- Create custom job and abstract roles.
- View the roles assigned to a user.
- View the users who have a specific role.

A user with the IT Security Manager or Application Implementation Consultant job role performs the Manage Job Roles task.

Manage Duties

The Oracle ERP Cloud and Oracle SCM Cloud security reference implementations provide many predefined duty roles. You can perform the Manage Duties task to:

- Review the duties of a job or abstract role.
• Manage the duties of a custom job or abstract role.
• Create custom duty roles.

A user with the IT Security Manager job role performs the Manage Duties task.

Define Data Security

You can manage Oracle ERP Cloud and Oracle SCM Cloud application data by performing tasks in these task lists:

• Define Data Security for Financials
• Define Data Security for Procurement
• Define Data Security for Product Management
• Define Data Security for Project Financial Management
• Define Data Security for Supply Chain Management

You can perform the tasks in these task lists to:

• Manage data access sets that secure ledgers.
• Manage Human Capital Management (HCM) security profiles that facilitate data role assignment for application users.
• Manage data security policies that determine grants of entitlement to a user or role on an object or attribute group.

A user with the IT Security Manager job role performs the tasks in the Define Data Security task lists.

Manage Role Provisioning Rules

You create role mappings to control the provisioning of all types of roles to application users by performing this task. For example, you can create a role mapping to provision the Accounts Payable Specialist role automatically to users that meet criteria specific to accounts payable users.

A user with the IT Security Manager job role performs the Manage Role Provisioning Rules task.

Manage Oracle Social Network Objects

You can determine which business objects in Oracle ERP Cloud are available for social collaboration by performing this task. For example, use this task to enable discussion among Oracle Social Network users about requisitions, purchase orders, invoices, payments, receipts, and other transactions.

A user with the IT Security Manager or Application Implementation Consultant job role performs the Manage Oracle Social Network Objects task.

Note: You can perform this task after you set up and configure Oracle Social Network. If you don’t use Oracle Social Network, you can skip this task.
6 Define Approval Management

Overview of Approval Management

Use approval management to define policies that apply to approval workflows. For example, to reflect your own corporate policies, you can specify levels of approval for transactions over a particular amount and determine how the approvals are routed.

Approval management:
- Controls workflows for business objects such as expense reports.
- Enables you to define complex, multistage task routing rules.
- Integrates with the setup in Human Capital Management (HCM) to derive approvers based on the supervisory hierarchy.

To define approval management:
- In the Offerings work area, enable the Approval Routing Administration feature at the offering level so that relevant setup tasks are available.
- In the Setup and Maintenance work area, use the following setup tasks in the Application Extensions or another functional area.
  - Manage Task Configurations
  - Manage Approval Groups

Task Configuration

Manage rule sets and rules that control approval flows.
- To configure a predefined approval policy, select the predefined rule set and click the Edit Task icon.
- To disable a predefined rule set, select the Ignore participant check box for that rule set.
- To edit the rules within a predefined rule set, you can insert, update, or delete while in edit mode.
- You can configure a specific rule to automatically approve a task without sending it to any approver.
  - Modify the routing for that rule so that it is sent to the initiator (which means the requestor is the approver).
  - Set the Auto Action Enabled option to True.
  - Enter APPROVE in the Auto Action field.

Approval Groups

Each approval group includes a set of users that you configure to act on tasks in a certain pattern. Tasks can be defined to get routed to an approval group instead of an individual user.
- You can nest approval groups within approval groups.
- You have two options for defining the group:
### Access for Workflow Administrators

Predefined *roles* provide *workflow* administration access for specific product families. These roles are assigned by default to predefined *job roles*. Administrators with these roles can, for example, set up approval rules and manage submitted approval tasks for corresponding product families. One predefined role gives access for all families and isn't assigned by default to any predefined job role.

#### Predefined Roles

This table lists the predefined roles for workflow administration access and the predefined job roles that they’re assigned to.

<table>
<thead>
<tr>
<th>Product Family</th>
<th>Role Name</th>
<th>Role Code</th>
<th>Predefined Job Roles Assigned To</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>BPM Workflow All Domains Administrator Role</td>
<td>BPMWorkflowAllDomainsAdmin</td>
<td>None</td>
</tr>
<tr>
<td>Financials</td>
<td>BPM Workflow Financials Administrator</td>
<td>BPMWorkflowFINAdmin</td>
<td>Financial Application Administrator (ORA_FUN_FINANCIAL_APPLICATION_ADMINISTRATOR_JOB)</td>
</tr>
<tr>
<td>Higher Education</td>
<td>BPM Workflow Higher Education Administrator</td>
<td>BPMWorkflowHEDAdmin</td>
<td>Higher Education Application Administrator (ORA_HEY_HIGHER_EDUCATION_APPLICATION_ADMINISTRATOR_JOB)</td>
</tr>
<tr>
<td>Human Capital Management</td>
<td>BPM Workflow Human Capital Management</td>
<td>BPMWorkflowHCMAadmin</td>
<td>Human Capital Management Application Administrator (ORA_HRC_HUMAN_CAPITAL_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB)</td>
</tr>
<tr>
<td>Incentive Compensation</td>
<td>BPM Workflow Incentive Compensation Administrator</td>
<td>BPMWorkflowOICAdmin</td>
<td>Customer Relationship Management Application Administrator (ORA_ZCA_CUSTOMER_RELATIONSHIP_MANAGEMENT_...)</td>
</tr>
</tbody>
</table>

---

- Static: Select the specific users to include in the group.
- Dynamic: Provide the logic to use to determine the users in the group.

### Related Topics

- Configure Offerings
- Update Existing Setup Data
### Oracle ERP Cloud
Implementing Common Features for Financials and Project Portfolio Management

#### Chapter 6
Define Approval Management

<table>
<thead>
<tr>
<th>Product Family</th>
<th>Role Name</th>
<th>Role Code</th>
<th>Predefined Job Roles Assigned To</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>APPLICATION_ADMINISTRATOR_JOB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Incentive Compensation Application Administrator (ORA_CN_INCENTIVE_COMPENSATION_ADMINISTRATOR_JOB)</td>
</tr>
<tr>
<td>Procurement</td>
<td>BPM Workflow Procurement Administrator</td>
<td>BPMWorkflowPRCAAdmin</td>
<td>Procurement Application Administrator (ORA_PO_PROCUREMENT_APPLICATION_ADMINISTRATOR_JOB)</td>
</tr>
<tr>
<td>Project Portfolio Management</td>
<td>BPM Workflow Project Administrator</td>
<td>BPMWorkflowPRJAdmin</td>
<td>Project Application Administrator (ORA_PJF_PROJECTS_APPLICATION_ADMINISTRATOR_JOB)</td>
</tr>
<tr>
<td>Sales</td>
<td>BPM Workflow Customer Relationship Management Administrator</td>
<td>BPMWorkflowCRMAadmin</td>
<td>Corporate Marketing Manager (ORA_MKT_CORPORATE_MARKETING_MANAGER_JOB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Customer Relationship Management Application Administrator (ORA_ZCA_CUSTOMER_RELATIONSHIP_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Marketing Analyst (ORA_MKT_MARKETING_ANALYST_JOB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Marketing Manager (ORA_MKT_MARKETING_MANAGER_JOB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Marketing Operations Manager (ORA_MKT_MARKETING_OPERATIONS_MANAGER_JOB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Marketing VP (ORA_MKT_MARKETING_VP_JOB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Sales Lead Qualifier (ORA_MKL_SALES_LEAD_QUALIFIER_JOB)</td>
</tr>
<tr>
<td>Supply Chain Management</td>
<td>BPM Workflow Supply Chain Administrator</td>
<td>BPMWorkflowSCMAdmin</td>
<td>Supply Chain Application Administrator (ORA_RCS_SUPPLYCHAIN_ADMINISTRATOR_JOB)</td>
</tr>
</tbody>
</table>
Usage of the Roles

If your administrators manage workflow for more than one product family, then you or your security administrator can add the appropriate family-specific roles to custom roles for those users. If your administrators manage workflow for all product families, then add BPM Workflow All Domains Administrator Role to a custom role for those users.

Note:
- Assign BPM Workflow All Domains Administrator Role only if your administrators truly need access for all product families. For multiple product families, but not all, assign instead the roles for the corresponding families.
- To-do tasks are visible to all administrators no matter which role they have for workflow administration access.

Related Topics
- Assign Roles to an Existing User
- Edit Job and Abstract Roles
- Role Copying or Editing

Define the Due Date and Expiration Policies for Workflow Tasks

For workflow tasks that should be completed within a general time frame, you can set a due date, expiration policies, or both. The current assignee will get notified before the due date to take action. Even after the due date passes, the task doesn't expire and the assignee, as well as any subsequent approvers, can still act on it. But if you set expiration policies, the task can expire based on your settings. Expired tasks are in a final state and no one can make any more updates to them. To set due dates and expiration policies, use the Manage Task Configurations or other approval setup task in the Setup and Maintenance work area.

Overall Process

To set the due date, expiration policies, or both:

1. In the Setup and Maintenance work area, go to the following:
   - Functional Area: Application Extensions or another functional area
   - Task: Manage Task Configurations or another approval setup task
2. In BPM Worklist, on the Task Configuration tab, select the workflow task to configure and click the Edit task icon in the Tasks to be configured toolbar.
3. Open the Deadlines subtab and make your changes.
4. Click the Commit task icon in the Tasks to be configured toolbar when you're ready to deploy your changes.
Specify a Due Date

On the Deadlines subtab, specify the time frame that all approvers should complete the task in. For example, if you enter 14 days for the **Due Date** fields, that means the task is due 14 days after it's created.

Indicate How Tasks Expire

Follow these steps:

1. On the Deadlines subtab, expand the Expiration Settings section.
2. Indicate how the task would expire, based on what's not done during a specified time frame.
   - **Task Level**: If all approvals aren't done.
   - **Assignee Level**: If the assignee doesn't act on the task.
Define Policies with Assignee Level Expiration

To set expiration policies, including any escalations or renewals, with Assignee Level selected:

1. In the Expiration Settings section of the Deadlines subtab, specify a duration and optionally select the Exclude Saturday and Sunday check box. For example, if you enter 30 days and select the check box, then:
   - For sequential routing, the task expires if the last assignee doesn't act on the task within 30 weekdays after the task is routed to that assignee. If the first assignee doesn't act in 30 weekdays, the task is passed to the next assignee, who gets another 30 weekdays. And so on, until the last assignee.
   - For parallel routing, the task expires if the current assignees don't act on the task within 30 weekdays after the task is assigned.

2. If you want tasks to get escalated or renewed after they expire, select the Escalate or Renew option. Otherwise, leave the Expire only option selected.

3. To escalate, indicate how many times the approval escalates up the management chain. For example, you enter 2 in the Maximum Escalation Levels field. When the task expires:
   - For sequential routing, the task is routed to the manager (User 2) of the last assignee (User 1).
   - For parallel routing, the task is routed to the managers (User 2) of all current assignees (User 1).

When User 2 doesn't act within 30 weekdays, then the task is escalated to the manager of User 2, who has another 30 weekdays before the task goes into a final Expired status.

4. To renew, indicate how many times the task can get renewed. For example, you enter 2 in the Maximum Renewals field. When the task expires:
   - For sequential routing, the last assignee gets another 30 weekdays.
   - For parallel routing, the current assignees get another 30 weekdays.

Define Policies for Task Level Expiration

To set expiration policies, including any escalations or renewals, with Task Level selected:

1. In the Expiration Settings section of the Deadlines subtab, specify a duration and optionally select the Exclude Saturday and Sunday check box. For example, if you enter 30 days and select the check box, then the task expires if not all approvals are done 30 weekdays after the task is routed to the first assignee. If there are three assignees and the first two take 25 weekdays to act, then the last assignee only gets 5 weekdays.

2. If you want tasks to get escalated or renewed after they expire, select the Escalate or Renew option. Otherwise, leave the Expire only option selected.

3. To escalate, indicate how many times the approval escalates up the management chain. For example, you enter 2 in the Maximum Escalation Levels field and select Director in the Highest Approver Title field. When the task expires, it's routed to the manager (User 2) of the current assignee (User 1). When User 2 doesn't act within 30 weekdays, then:
   - If User 2 isn't a director, the task is escalated to the manager of User 2, who has another 30 weekdays before the task goes into a final Expired status.
   - If User 2 is a director, then the task goes into a final Expired status.

4. To renew, indicate how many times the task can get renewed. For example, you enter 2 in the Maximum Renewals field. When the task expires, all pending assignees get another collective 30 weekdays to act. If they don't all act within that period, then the task is renewed for another 30 weekdays. If the task still isn't complete in that time, then it goes into a final Expired status.
Define People to Support Workflow Tasks

Generally, workflow tasks involve the person who creates the task and the approvers who act on the task. But for any given task, you can also define others who might get involved:

- **Task Owner**: The task owner is an administrator for the business process that the workflow task is related to. Task owners can see the tasks for their business processes and act on behalf of anyone assigned to the task. They can also reassign, withdraw, or escalate tasks.

- **Reviewers**: Reviewers can add comments and attachments to a task without having the task directly assigned to them. They can do this only if you or someone else set them up as reviewers for the task.

- **Error Assignees**: Sometimes workflow tasks run into problems when trying to figure out the next assignee, for example when trying to carry out the escalation policy. You can define whom to automatically assign the task to so that the issue gets fixed. You can have different error assignees for different tasks. Error assignees can route or reassign the task to the appropriate people, or indicate that the issue can't be fixed (in which case, the task is set to the Error status).

Set Up Task Owner, Reviewers, or Error Assignees

This screenshot shows where you define supporting people on the Task Configuration tab in BPM Worklist.

1. In the Setup and Maintenance work area, go to the Manage Task Configurations task or another approval setup task in the Application Extensions functional area or another functional area.
2. In BPM Worklist, on the Task Configuration tab, select the workflow task.
3. Click the **Edit task** icon in the **Tasks to be configured** toolbar.
4. Open the Assignees subtab.
5. Click one of the icons after the **Task Owner**, **Reviewers**, or **Error assignees** field.
   - First icon: Use the expression builder to define who gets assigned, and click **OK**.
   - Second icon: Find the person you want, and click **OK**.
6. Click the **Commit task** icon in the **Tasks to be configured** toolbar when you’re ready to roll out your setup.
Assignments and Routing

Prevent Assigning Approvals to Specific Users

In rare cases, just based on how approval rules are set up, workflow tasks get routed to the person who created the task. Or, to someone else who should not be approving due to a conflict of interest. To make sure that such things don't happen, you can configure tasks so that they skip certain users in the approval chain. Those users can still get FYI notifications about the tasks, but not notifications they can act on.

1. In the Setup and Maintenance work area, go to the Manage Task Configurations task or another approval setup task in the Application Extensions functional area or another functional area.
2. In BPM Worklist, on the Task Configuration tab, select the workflow task.
3. Click the Edit task icon in the Tasks to be configured toolbar.
4. Click the Configuration subtab.
5. In the Prohibit User Self-Approval section, select the Prohibit self-approval by users named in these payload attributes check box.
6. In the Payload Attributes subsection, add one or more attributes:
   a. Click the Add icon.
   b. Click the Expression Builder icon.
   c. In the Expression Builder, expand Variables > Task > task:task.
   d. Select an attribute that represents the users to skip, for example task:creator for the task creator. Or, open the task:payload node and select an attribute from there instead.
   e. Click Insert into Expression and then OK.
7. Select the Reassign approvals to those users' managers check box if you want to reroute tasks to the manager of the skipped user. Otherwise, the task goes to the next assignee in the approval chain.
8. In the Tasks to be configured toolbar, click the Commit task icon when you're ready to roll out your changes.

You can also do this setup for specific participants in the task, if it's not an FYI task. In the Assignees subtab, select the participant, click Advanced, and find the same Prohibit User Self-Approval section.

- If you have settings at both the participant and task level, both would apply.
- If there's any conflict, for example with the setting of the Reassign approvals to those users' managers check box, the participant level setting takes precedence.

How You Can Regularly Reassign Pending Approvals for Workers That Become Inactive

A manager assignment can become inactive due to the end of an assignment or work term, termination, or global transfer. If the manager has any pending approval notifications, you must reassign them.
Run this process: Run Reassign Pending Approvals for Terminations and Correct Invalid Supervisor Assignments Process in the Scheduled Processes work area. You can set a schedule to run it at least once a day. You can run it more frequently if you want things updated faster.

Here’s what the process does:

- It reassigns the direct reports of a terminated manager to that person’s line manager and also assigns any pending notifications to the line manager. Only actionable notifications will be reassigned.
- It reassigns pending approval notifications based on the number of days you specify using the Past Period in Days Considered for Reassigning Pending Approvals parameter.

**Related Topics**

- Submit Scheduled Processes and Process Sets

---

**Workflow Notifications**

**Disable or Enable Workflow Notifications**

When workflow tasks are assigned to users, they get notifications through email and the Notifications icon in the global header. Workflow tasks are managed in the Worklist: Notifications and Approvals work area and configured in the Setup and Maintenance work area using the Manage Task Configurations or other approval setup task. If you have the BPM Workflow System Admin Role (BPMWorkflowAdmin) role, you can disable or enable these notifications for all users. For example, you can disable notifications during testing, to avoid sending test notifications to users, and then enable notifications when ready.

When you disable workflow notifications:

- The setting applies only to email notifications that are sent as part of workflow tasks, not to all emails in general.
- Users can still find their workflow tasks in the Worklist: Notifications and Approvals work area.

**Setting Notification Mode**

To disable or enable workflow notifications:

1. Click the **Notifications** icon on the global header.
2. Click **More Details**.
3. In BPM Worklist, click your user name and select **Administration**.
4. On the Application Preferences page that’s on the Administration tab, select a value from the **Notification Mode** list:
   - **All**: Email notifications are enabled. Workflow notifications are included in the global header. This is the default value.
   - **None**: Email notifications and workflow notifications in the global header are disabled.
   - **Email**: Only email notifications are enabled. New workflow notifications won’t appear in the global header.
   - **In-app**: Workflow notifications in the global header are enabled. Email notifications are no longer sent.
Note: If you or another administrator has selected news feed as the default home page layout, then users also get notifications in the Things to Finish section on the home page, as well as the Notifications page. The same mode setting that applies to the notifications in the global header also applies to the Things to Finish section and the Notifications page.

5. Click **Save**.

Define When to Send Workflow Notifications

When notifications are enabled, each workflow task is set up by default to send notifications as part of the approval process. For example, to notify assignees whenever they're assigned a task. For any workflow task, you can change the predefined setup to determine when notifications are sent, and to whom. This setup applies to email or in-app notifications, or both, depending on what's enabled.

Set Up Notification Scenarios

To define the scenarios for sending notifications:

1. Navigate to the Notifications subtab in BPM Worklist for the task you want to edit.
2. Click the **Add Notification** icon to enable additional notification scenarios, or edit existing rows directly.
   a. In the **Task Status** column, select when to send the notification, for example when the task has expired. Aside from the actions or statuses available to end users, you can also select any of the following:
      - **Alerted**: Usually, an error condition that can be fixed. The task is assigned to the error assignee, or someone else if the task doesn't have error assignees.
      - **Update**: Whenever the task is updated, for example by adding a comment or attachment, without affecting the approval status or routing.
      - **Update Outcome**: Whenever the outcome of the task is updated, for example approved or rejected.
      - **All other actions**: Any action that's not already in the list of values.
   b. In the **Recipient** column, select whom to notify.
      - **Assignees**: The users or groups whom the task is currently assigned to.
      - **Initiator**: The user who created the task.
      - **Approvers**: The users who already approved the task as part of a sequential approval chain.
      - **Owner**: The task owner, who's responsible for the business process that the task is related to.
      - **Reviewer**: The user who can only add comments and attachments to a task.
3. To disable specific notification scenarios, select a row and click the **Delete Notification** icon.
4. Click the **Save** icon in the **Tasks to be configured** toolbar.
The following figure shows the table on the Notifications subtab with predefined scenarios for a workflow task. In this example, notifications are sent to assignees whenever the task is assigned to them. The task initiator also gets a notification when the task is complete, and administrators are notified if the task results in error.

<table>
<thead>
<tr>
<th>Task Status</th>
<th>Recipient</th>
<th>Notification Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign</td>
<td>Assignees</td>
<td></td>
</tr>
<tr>
<td>Complete</td>
<td>Initiator</td>
<td></td>
</tr>
<tr>
<td>Error</td>
<td>Owner</td>
<td></td>
</tr>
</tbody>
</table>

Set Up Reminders

To send reminders in addition to the defined notification scenarios:

1. Select the Enable Reminder check box.
2. From the Repeat list, select the number of reminders to send, for example 2.
3. From the Initiating Action list, specify if the reminder is sent based on when the task is assigned to a user or when the task expires.
4. Define a frequency for the time between reminders, for example 3 days.
5. Click the Save icon in the Tasks to be configured toolbar.

The following figure shows the reminder setup with the given sample settings, along with After Assignment selected as the initiating action. For this example, a reminder is sent three days after the user is assigned the task. One more reminder is sent three days after that, if the user still hasn't acted on the assigned task.

Related Topics
- Actions and Statuses for Workflow Tasks
Synchronize Notifications in the Global Header with Workflow Tasks

When workflow tasks are assigned to users, they get an email as well as a notification in the global header. They can also find all of their workflow tasks in the Worklist: Notifications and Approvals work area. The notifications in the global header don't immediately reflect changes to the task status due to actions taken through email, the Worklist: Notifications and Approvals work area, or BPM Worklist. Use the Synchronize Notifications in Global Header scheduled process to update the notifications with the latest task statuses, which are always reflected in the Worklist: Notifications and Approvals work area.

Note: If you or another administrator has selected news feed as the default home page layout, then users also get notifications in the Things to Finish section on the home page, as well as the Notifications page. The scheduled process also applies to notifications in these UIs. For example, the Things to Finish section automatically reflects changes made in the global header, but not changes made through email until the scheduled process runs.

Scheduling the Process

In the Scheduled Processes work area, submit the Synchronize Notifications in Global Header process with a defined schedule. For example, schedule the process to run every two hours.

Effects of the Synchronization

After the scheduled process runs, notifications in the global header might move from the Pending Notifications list to the All Notifications list. For example:

1. A notification is pending a user’s approval.
2. The user approves the task using the Worklist: Notifications and Approvals work area. The task status changes, but the notification in the global header is still in the Pending Notifications list.
3. After synchronization, the notification moves to the All Notifications list because the user has changed the task status to Approved, and the notification is no longer pending action.

If the news feed home page layout is selected, then after synchronization, the notification:

- Is removed from the list in the global header
- Is no longer a card in the Things to Finish section
- Moves from the Assigned to Me tab on the Notifications page to the All tab

The scheduled process doesn’t update the title of notifications in the global header. Similar to email subjects, the notification titles are static.

Related Topics

- Submit Scheduled Processes and Process Sets
Email Notifications

Add Header Content to Workflow Email Notifications
Each workflow task is configured with scenarios for sending email notifications as part of the approval process. For each notification scenario in the Notifications subtab, the Notification Header column determines what's in the email header, a region that appears before the email body.

- By default, all predefined notification scenarios have emails with blank headers.
- Any notification scenarios you add in the Notifications subtab would have the following header value: `concat(string('Task '), /task:task/task:title, string(' requires your attention.'))`. It is recommended to change that value to `null`.

For some workflow tasks, you can enable configurable email notifications based on report layouts to be used instead of the standard email notifications. The Notification Header setting doesn't apply to those configurable email notifications.

Adding Company Name or Logo
If you do want to add, for example, your company name or logo to the email header:

1. In the Setup and Maintenance work area, go to the Manage Task Configurations task or another approval setup task in the Application Extensions functional area or another functional area.
2. In BPM Worklist, on the Task Configuration tab, select the workflow task.
3. Click the Edit task icon in the Tasks to be configured toolbar.
4. Open the Notifications subtab.
5. For the specific notification scenario on the Notifications subtab, click the icon in the Notification Header column.
6. In the Edit Notification Message dialog box, delete any existing content and enter the following in the Notification Message field.
   - For company name: Enter text in single quotes, for example ‘Oracle’. You can also use HTML formatting, for example ‘<h2>Oracle</h2>’.
   - For company logo: Enter the URL to your logo, following this syntax: `'<img src="https://cloud.oracle.com/res/images/header/oracle-cloud-logo.png" width="230" height="69" alt="Oracle Logo">'`. Replace the URL and alternative text with your own.
7. Click the Save icon in the Tasks to be configured toolbar.

Set Up the From Field in Workflow Email Notifications
By default, the From field in workflow email notifications shows an email address without a sender name. You can't change the email address, but you can specify the sender name. For example, if you indicate that Your Company is the text to display, then the From field shows: `Your Company <<your pod>.fa.sender@workflow.mail.<your data center>.cloud.oracle.com>`.

You can set up the sender name in application preferences for all workflow tasks, or have different setup for specific workflow tasks. If not specified at the task level, the sender name setting defaults from the preferences.
Setting Up for All Workflow Tasks
To define the sender name for all workflow tasks that have no other applicable setup:

1. Open the Administration tab. If you’re not in BPM Worklist:
   a. Click the **Notifications** icon in the *global header*.
   b. Click **More Details**.
   c. Click your user name and select **Administration**.
2. On the Application Preferences page that’s on the Administration tab, select one of the **Email "From:" Display Name** options.
   a. Select to specify the text to display. Enter your value or leave blank if you want nothing to appear in the From field.
   b. Select **Submitter** to show the person who created the task.
   c. Select **Previous Approver** to show the previous assignee in the approval chain. When the notification is sent to the first assignee in the approval chain, the From field shows the person who created the task.
3. Click **Save**.
Setting Up for a Specific Workflow Task
To specify the sender name for a specific workflow task:

1. In the Setup and Maintenance work area, go to the Manage Task Configurations task or another approval setup task in the Application Extensions functional area or another functional area.
2. In BPM Worklist, on the Task Configuration tab, select the workflow task.
3. Click the Edit task icon in the Tasks to be configured toolbar.
4. Open the Notifications subtab.
5. On the Notifications subtab, click the Expand More icon.
6. Select one of the Email "From:" Display Name options.
   - Select Not Applicable so that what appears in the From field depends on the application preferences that apply to all workflow tasks.
   - Select to specify the text to display. Enter your value in quotes, for example "Oracle", or leave blank if you want nothing to appear in the From field.
   - Select Previous Approver to show the previous assignee in the approval chain. When the notification is sent to the first assignee in the approval chain, the From field shows the person who created the task.
7. Click the Save icon in the Tasks to be configured toolbar.

Send Test Workflow Email Notifications to One Address
While you're testing workflow setup, you can send all email notifications to a single address so that your users don't receive any test emails. The test emails are still sent based on the notification scenarios defined for the particular workflow task.

Specifying the Email Address
To enter the email address to send test emails to:

1. Click the Notifications icon in the global header.
2. Click More Details.
3. Click your user name and select Administration.
4. On the Application Preferences page that's on the Administration tab, click the Test Notification Email Address icon.
5. In the dialog box, enter an email address in the Test Notification Email Address field.
6. Click OK and then Save.

After you’re done testing, go back and delete the email address that you entered.

More Setup for Workflow Email Notifications
You can click the Expand More icon on the Notifications subtab to open the More section and see other setup options for email notifications. In general, leave the default settings in this section for every workflow task. Settings in this section include the following check boxes, which, if selected, would:

- **Make notification secure (exclude details):** Exclude business transaction details in email notifications.
- **Hide End User Web URL in notifications:** Remove the default first line in the email body: Access this task in the Workspace Application or take direct action using the links in this email. This line includes a link that opens BPM Worklist. It is recommended to select this check box.
- **Make notification actionable:** Include links in email notifications that users can click to directly take action, for example to approve or reject.
- **Send task attachments with email notifications:** Include files attached to the task as attachments in the email notifications.

Conﬁgurable Notiﬁcations

Overview of Financials Conﬁgurable Workflow Notiﬁcations
The Financial applications, as part of certain business ﬂows, automatically send notiﬁcations for review or approval. For example, when a user submits an expense report, the approvers receive an email containing the approval request. Oracle Business Intelligence (BI) Publisher reports are used for some ﬂows to generate the content and format. You can enable BI Publisher-based notiﬁcations, which are ready to use as delivered. The notiﬁcation templates can be easily conﬁgured to meet other speciﬁc requirements. If required, you can change the delivered template layouts and content, to add images, change colors and styling, add or remove attributes or modify text.

The following table shows the ﬁnancial products that use conﬁgurable notiﬁcations, along with their associated features, and workflow task names:

<table>
<thead>
<tr>
<th>Product</th>
<th>Feature Name</th>
<th>Report or Notification Name</th>
<th>Workflow Task Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Ledger</td>
<td>Journal Approval Notifications</td>
<td>Journal Approval Report</td>
<td>FinGlJournalApproval</td>
</tr>
<tr>
<td>Payables</td>
<td>Invoice Approval Notifications</td>
<td>Invoice Approval Report</td>
<td>FinApInvoiceApproval</td>
</tr>
<tr>
<td>Payables</td>
<td>Hold Resolution Notifications</td>
<td>Hold Resolution Notification</td>
<td>FinApHoldApproval</td>
</tr>
<tr>
<td>Payables</td>
<td>Payment Approval Notifications</td>
<td>Payment Approval Notification</td>
<td>PaymentApproval</td>
</tr>
<tr>
<td>Product</td>
<td>Feature Name</td>
<td>Report or Notification Name</td>
<td>Workflow Task Name</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>Payables</td>
<td>Invoice Account Coding Notifications</td>
<td>Invoice Account Coding Email Notification</td>
<td>FinApInvoiceAccountCoding</td>
</tr>
<tr>
<td>Expenses</td>
<td>Expense Report Approval Notifications</td>
<td>Expense Report Approval Notification</td>
<td>FinExmWorkflowExpenseApproval</td>
</tr>
<tr>
<td>Expenses</td>
<td>Expense Reimbursement Notifications</td>
<td>Reimbursement Paid to Card Issuer Notification</td>
<td>FinExmReimToCardIssuerFyi</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reimbursement Paid by Check Notification</td>
<td>FinExmReimToEmpByCheckFyi</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reimbursement Paid by Direct Deposit Notification</td>
<td>FinExmReimToEmpByDepositFyi</td>
</tr>
<tr>
<td>Expenses</td>
<td>Expense Audit and Receipt Management Notifications</td>
<td>Expense Report Adjusted by Auditor Notification</td>
<td>FinExmExpenseAuditFyi</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Expense Report Rejected by Auditor Notification</td>
<td>FinExmExpenseAuditFyi</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pending Payment with Warnings Notification</td>
<td>FinExmReceiptManagementFyi</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Expense Report Returned by Auditor Notification</td>
<td>FinExmReportShortpaidReceiptIssue</td>
</tr>
<tr>
<td>Expenses</td>
<td>Cash Advance Notifications</td>
<td>Cash Advances Approval Notification</td>
<td>FinExmWorkflowCashAdvanceApproval</td>
</tr>
<tr>
<td>Expenses</td>
<td>Expense Audit and Receipt Management Notifications</td>
<td>Payment is Held Notification</td>
<td>FinExmReceiptManagementFyi</td>
</tr>
<tr>
<td>Expenses</td>
<td>Expense Audit and Receipt Management Notifications</td>
<td>Payment is Released Notification</td>
<td>FinExmReceiptManagementFyi</td>
</tr>
<tr>
<td>Expenses</td>
<td>Expense Reimbursement Notifications</td>
<td>Reimbursement Has Been Paid to Employer Notification</td>
<td>FinExmReimToEmpByCheckFyi</td>
</tr>
<tr>
<td>Expenses</td>
<td>Expense Audit and Receipt Management Notifications</td>
<td>Mileage Adjustment Notification</td>
<td>ExpenseAuditMileageAdjustmentFyi</td>
</tr>
<tr>
<td>Expenses</td>
<td>Expense Audit and Receipt Management Notifications</td>
<td>Missing Receipt Declaration is Required Notification</td>
<td>FinExmReceiptManagementFyi</td>
</tr>
<tr>
<td>Product</td>
<td>Feature Name</td>
<td>Report or Notification Name</td>
<td>Workflow Task Name</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------</td>
<td>------------------------------------------------------</td>
<td>---------------------------------------------------------</td>
</tr>
<tr>
<td>Expenses</td>
<td>Cash Advances Notifications</td>
<td>Overdue Cash Advance Notification</td>
<td>FinExmWorkflowOverdueNotification</td>
</tr>
<tr>
<td>Expenses</td>
<td>Cash Advances Notifications</td>
<td>Cash Advances Automatically Applied Notification</td>
<td>FinExmWorkflowCashAdvanceAutoAppliedFyi</td>
</tr>
<tr>
<td>Expenses</td>
<td>Expense Audit and Receipt Management Notifications</td>
<td>Inactive Employee's Outstanding Transactions Notification</td>
<td>FinExmInactiveEmployeeSubmitExpenseFyi</td>
</tr>
<tr>
<td>Expenses</td>
<td>Expense Audit and Receipt Management Notifications</td>
<td>Inactive Employee-Accept Responsibility Notification</td>
<td>FinExmInactiveEmployeeAcceptResp</td>
</tr>
<tr>
<td>Intercompany</td>
<td>Intercompany Email Notifications</td>
<td>Intercompany Transaction Approval Notification</td>
<td>FinFunTransactionApproval for Intercompany Transaction Approval Notification</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Intercompany Enter Receiver Distribution Notification</td>
<td>FinFunEnterDistRequestForAction for Intercompany Enter Receiver Distribution Notification</td>
</tr>
<tr>
<td>Bill Management</td>
<td>User Registrations Notifications</td>
<td>Registration Confirmation.</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>New Account Registration Confirmation.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Account Access Revocation Notification.</td>
<td></td>
</tr>
<tr>
<td>Budgetary Control</td>
<td>Budgetary Control Override Email Notifications</td>
<td>Budget Override Request Notification Budget</td>
<td>OverrideNotificationFyi</td>
</tr>
<tr>
<td>Budgetary Control</td>
<td>Budgetary Control Override Email Notifications</td>
<td>Override Taken Notification</td>
<td>OverrideNotificationFyi</td>
</tr>
<tr>
<td>Receivables</td>
<td>Manual Credit Memo Request Notification</td>
<td>Manual Credit Memo Request Notification</td>
<td>FinArTrxsCreditMemosCreationPostProcessing</td>
</tr>
</tbody>
</table>

**Note:** The in-app notification is also configurable.
Note: When you receive a workflow email notification, you can alternatively find the same notification by clicking the Notifications icon in the global header or opening the Worklist: Notifications and Approvals work area. Your report edits don’t affect the other notification methods.

Process Overview
Generating configurable notifications through BI Publisher involves various types of objects in the BI catalog, including data models, subtemplates, style templates, and reports. Reports pull data from data models and generate notifications in an HTML format. The report layout templates use common table and paragraph styles and refer to a central subtemplate that contains reusable notification components. This figure shows how these BI objects work together to generate the notification content.

- **Data Sources**: Store the attributes and attribute values for business objects and transactions in the application (example of data sources being transaction tables)
- **Data Model**: Determines which attributes from data sources are available to be included in the notification and how that data is retrieved
- **Subtemplate**: Provides common components, such as a branding logo and buttons, that can be reused in multiple reports.
- **Style Template**: Provides styles such as the type of lines and fonts to use in tables, or the font type, size, and color to use for headings
- **Report**: Contains a layout template that determines:
  - Which attributes appear in the notification, from the data model used for the report
  - What the notification looks like, leveraging components from the subtemplate and styles from the style template used for the report
- **HTML**: Format of the output generated by the report
- **Email**: What the notification looks like, leveraging components from the subtemplate and styles from the style template used for the report
- In-App Notification: Has the HTML output embedded in the application UI
Each workflow task with configurable notifications has a corresponding predefined report in the BI catalog. For example, the Invoice Approval report contains the Invoice Approval Notifications report layout template and uses the Invoice Approval Data Model.

Notification Modifications

When you enable configurable email notifications, the predefined reports and related objects in the BI catalog will be used to generate the email notifications. The report-based notifications provide the same information as the standard notifications. In addition, the format of report-based notifications is optimized for mobile devices. To modify the email notifications, you can edit copies of the predefined reports, data models, and subtemplate. However, you can’t change the style template. You proceed as you would to edit any report, data model, or subtemplate in the catalog, for example:

1. Find a predefined report for expense approval in the BI catalog.
2. Use the Customize option to create a copy, or copy the report and paste it within the Custom folder.
3. Edit the copied report layout template.

Before modifying workflow email notifications, it's recommended that you familiarize yourself with BI Publisher in general. This will improve your ability to change your notifications by formatting emails to meet your business requirements.

With BI Publisher-based email notifications you:

- Use only the Template Builder for Word add-in to edit the .rtf template in Microsoft Word, rather than the layout editor or other tools available for creating and editing report layout.
- Edit a copy of predefined layout templates, rather than creating reports or layout templates.

Note: Review the My Oracle Support (MOS) note: Configurable Workflow Notifications: Implementation Considerations (Doc ID 2215570.1).

Steps to Enable

In the Offerings work area, enable the Configurable Email Notifications feature as follows:

1. Offering: Financials
2. Functional Area: Budgetary Control, Expenses, Intercompany, General Ledger, or Payables
3. Feature: Configurable Email Notifications
4. Choices: The specific notifications you want to enable
5. Click Done.

Related Topics

- Example of Modifying Expenses Email Notifications Using Oracle Business Intelligence Publisher
- Example of Modifying Journal Approval Notifications Using Oracle Business Intelligence (BI) Publisher
- Example of Modifying Invoice Approval Workflow Notifications Using Oracle Business Intelligence (BI) Publisher

Apply Changes to Workflow Notifications Immediately After Upload

Configurable workflow notifications are refreshed every 24 hours so that they perform better for your users. But when you're making changes to reports, subtemplates, or data models, you can apply your changes immediately so they're available for testing. Create profile options to turn off the refresh so that notifications reflect your changes immediately after you upload them to the BI catalog, instead of after the next refresh. When you're done configuring notifications, use the same profile options to turn the refresh back on, to optimize performance.

Note: The refresh applies only to changes uploaded to the BI catalog. You can always preview changes to layout templates while you're editing in Microsoft Word.
Create Profile Options to Control the Refresh

Your profile options can apply to all workflow tasks, a product family, product, or single workflow task. Based on the scope you want, your profile option must have a profile option code that follows a certain format.

<table>
<thead>
<tr>
<th>Scope</th>
<th>Profile Option Code</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>BIP_CLIENT_DISABLE</td>
<td>BIP_CLIENT_DISABLE</td>
</tr>
<tr>
<td>Product Family</td>
<td>BIP_CLIENT_DISABLE_&lt;FAMILY&gt;</td>
<td>BIP_CLIENT_DISABLE_FIN</td>
</tr>
<tr>
<td>Product</td>
<td>BIP_CLIENT_DISABLE_&lt;FAMILY&gt;_&lt;PRODUCT&gt;</td>
<td>BIP_CLIENT_DISABLE_FIN_AP</td>
</tr>
<tr>
<td>Workflow Task</td>
<td>BIP_CLIENT_DISABLE_&lt;FAMILY&gt;<em>&lt;PRODUCT&gt;</em>&lt;WORKFLOW&gt;</td>
<td>BIP_CLIENT_DISABLE_FIN_AP_FINAPINVOICEAPPROVAL</td>
</tr>
</tbody>
</table>

The profile options with a smaller scope take precedence. For example, you have profile option A with a global scope and profile option B with a workflow task scope. You’re currently configuring notifications for that workflow task, so you use profile option B to turn the refresh off just for that task. But based on profile option A, the refresh is still on for all other configurable notifications in all product families. Profile option B takes precedence over profile option A for that one workflow task.

**Tip:** To find the product family or product code, go to the Setup and Maintenance work area, and then the Manage Taxonomy Hierarchy task in the Application Extensions functional area. In the hierarchy, expand the Oracle Fusion node and look in the Module Key column. To find the workflow task name, go to the Task Configuration tab in BPM Worklist and look in the Tasks to be configured pane.

Now you’re ready to create your profile options!

1. In the Setup and Maintenance work area, go to the Manage Applications Core Profile Options task in the Application Extensions functional area.
2. On the Manage Applications Core Profile Options page, click the New icon.
3. On the Create Profile Option page, enter the profile option code in the format that corresponds to the scope you want.
4. Enter a display name that you can easily remember to help you find the profile option later.
5. From the Application list, select Oracle Middleware Extensions for Applications.
6. From the Module list, select Application Core.
7. In the SQL Validation field, enter this:
   
   ```sql
   select meaning, lookup_code from fnd_lookups where lookup_type='FND_TRUE_FALSE_TYPE'
   ```

8. Click Save and Close.
9. On the Manage Applications Core Profile Options page, make sure your new profile option is selected in the Search Results: Profile Options subsection.
10. In the <Profile Option>: Profile Option Levels subsection, select the Enabled and Updatable check boxes for the Site level.
11. Save your work.
Disable or Enable the Refresh

In the Setup and Maintenance work area, use the Manage Applications Core Administrator Profile Values task in the Application Extensions functional area. Set your profile option at the Site level to True to turn off the refresh. When you’re done making and testing your changes, set the profile option to False to turn the refresh back on.

*Related Topics*

- Update Existing Setup Data
- Set Profile Option Values
- Modules in Application Taxonomy

**Best Practices for Layouts in Workflow Notifications**

Predefined workflow notifications based on report layout templates all follow a general format. When you edit a copy of these layout templates in Microsoft Word, follow the predefined layout as closely as possible for consistency. Also keep in mind shared components and mobile considerations.
General Structure

In general, the workflow notifications contain a set of components that are displayed in a certain order. The callouts in this figure identify the email notification components listed in the following table.

The callouts in this figure identify the in-app notification components listed in the following table. In addition to describing each component, the table also indicates if the component appears in the email notification, in-app notification, or both.
Callout | Component | Notification Type
--- | --- | ---
1 | Buttons with the primary actions to take on the task, such as **Approve** and **Reject**. These buttons aren’t part of the configurable, report-based notification content. | In-app
2 | Notification header listing key attributes of the workflow task and the associated transaction. | Both
3 | Buttons for the primary actions to take on the task, such as **Approve** and **Reject**. | Email
4 | Notification body that usually includes transaction and line level details, displayed in tables or sets of attributes with corresponding values. The data model for the report restricts the total number of rows displayed in some of the tables. If the limit is exceeded, the table footer provides a link to the transaction details page, where users can view all the rows. To change this limit, you can edit a copy of the data model. | Both
<table>
<thead>
<tr>
<th>Callout</th>
<th>Component</th>
<th>Notification Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Approval history, including any attachments that users in the history uploaded for the task. You can't edit the approval history component, which usually appears in the body of only email notifications. For in-app notifications, you can usually view the history by clicking the Actions button and selecting History.</td>
<td>Email (or both, in rare cases)</td>
</tr>
<tr>
<td>6</td>
<td>Buttons for the primary actions again.</td>
<td>Email</td>
</tr>
<tr>
<td>7</td>
<td>A link to the corresponding transaction page, and another link to the in-app notification.</td>
<td>Email</td>
</tr>
</tbody>
</table>

When you modify notifications, try to keep to this general structure and don’t remove essential elements such as the action buttons. Likewise, don’t change the styles in your layout template. The predefined style template should still apply to your notification; don’t edit a copy of the style template and apply that to your notification.

To add components to your notification, for example another table, consider first downloading another style template from My Oracle Support. This template contains Quick Parts content that you can use in Word when you do more advanced work on layout templates. For example, from the Quick Parts gallery, you can select and add the table that’s consistent in format with predefined tables already on your notification.

By default, the components that you add in the layout template appear in both email and in-app notifications, where available. You can add conditions to explicitly make a particular element, for example a field, appear only in one type of notification and not the other.

Shared Components
A predefined subtemplate in the business intelligence (BI) catalog applies to all predefined layout templates for workflow notifications. The subtemplate contains components that are shared among the notifications, for example:

- Branding logo, if you add one to the subtemplate, which would appear as the first component in the email body. The logo appears in email notifications only.
- Action buttons in email notifications.
- Links at the end of the email notification, one to the corresponding transaction page, and another to the in-app notification.

When you make a copy of a predefined layout template to edit, the copy automatically inherits the same predefined subtemplate. To edit these shared components, make a copy of the predefined subtemplate, edit the copied version, and apply it to your own layout templates.

Mobile Considerations
Because users can view the workflow notifications on mobile devices, always consider mobile first and keep the notifications as simple as possible. For example:
- Don’t put too much content horizontally, such as too many columns in tables.
- Keep all text, including attributes and column headings, as short as possible.
• Center align lists of attributes and their values, if they appear outside tables.

Make sure to test your email notifications on mobile devices.

Related Topics

• Subtemplates

• Define the Number of Rows in Tables

Add a Branding Logo and Change Other Shared Components in Workflow Notifications

A predefined subtemplate contains common components for all workflow notifications based on predefined report layouts. For example, the subtemplate has a place for you to add a branding logo, which would appear at the beginning of email notifications. You can modify other shared components so that the same changes apply to your notifications. For example, for email notifications, you can also change the text on action buttons, or the text of the links that appear at the end of emails.

Note:

• You must edit a copy of the subtemplate in the Custom folder of the business intelligence (BI) catalog. Don’t directly update the predefined subtemplate.

• The exact steps can vary depending on your version of Microsoft Word.

Modifying Shared Components in the Subtemplate

To edit a copy of the predefined subtemplate that contains the shared components:

1. Click Navigator > Reports and Analytics.
2. Click the Browse Catalog icon.
3. In the BI catalog (the Folders pane), expand Shared Folders > Common Content > Templates.
4. For Workflow Notification Subtemplate, click More and select Customize.

If you’re not using the Customize option:

a. Click Copy in the toolbar with Workflow Notification Subtemplate selected.
b. In the BI catalog, expand Shared Folders > Custom > Common Content > Templates. Create a Templates folder in this location if it doesn’t exist.
c. Click Paste in the toolbar.
d. Click the Edit link for the copied subtemplate.

All reports using the predefined subtemplate are automatically redirected to point to your subtemplate in the Custom folder. This applies:

• To all reports, predefined or not
• No matter if you copy and paste the subtemplate or use the Customize option
• Only if your subtemplate has the same name and relative file path within Custom as the predefined subtemplate

5. In the Templates section, click the link in the Locale column.
6. Save the subtemplate .rtf file to your computer.
7. Open the .rtf file with Microsoft Word.

• To add a logo, insert your own image in the subtemplate.
To change button or link text, edit the text accordingly. Make the same edits wherever that button or link text appears in the subtemplate.

**Caution:** To ensure that your layout templates reflect these changes without additional rework, don't edit any other text in the subtemplate .rtf file.

8. Update Word options to ensure that existing links remain intact in the subtemplate.
   a. Click File > Options > Advanced.
   b. In the Word Options dialog box, click Web Options in the General section.
   c. In the Web Options dialog box, open the Files tab.
   d. Deselect the **Update links on save** check box.

9. Save your changes in Word.

**Uploading the Modified Subtemplate**

To upload your subtemplate to the BI catalog:

1. In the BI catalog, expand **Shared Folders > Custom > Common Content > Templates**.
2. Click **Edit** for Workflow Notification Subtemplate.
3. In the Templates section, click the **Upload** icon.
4. Select your modified .rtf subtemplate and a locale, and click **OK** to overwrite the original subtemplate.

**Related Topics**

- Subtemplates
- How You Modify Copies of Predefined Reports
- Generate Sample Report Data

**Use Quick Parts for Workflow Notifications**

Use the Quick Parts feature in Microsoft Word to easily insert reusable pieces of formatted content. When you edit copies of predefined report layout templates for workflow notifications in Word, you can add predefined Quick Parts content to your .rtf file. For example, you can insert a table in a format that's consistent with predefined notifications. The predefined Quick Parts content is available in a style template .dotx file on My Oracle Support.

**Note:** The exact steps can vary depending on your version of Microsoft Word.

**Prerequisites**

To get the predefined Quick Parts content into your Quick Parts gallery:

2. Download the .dotx file and save it to your Microsoft Word template folder, for example C:\Users\<user name>\AppData\Roaming\Microsoft\Templates.

Also, to preview your layout template changes before uploading the .rtf file back to the business intelligence (BI) catalog:

- Generate sample report data from the data model for the report that you're editing.
- Download a local copy of the subtemplate that applies to the layout template.

**Adding Quick Parts Content to Workflow Notifications**

To insert content from the Quick Parts gallery into a layout template:

1. In the BI catalog, find the predefined report with the layout template that you want to modify.
2. For the report, click More and select Customize.
   If you’re not using the Customize option:
   a. Copy the predefined report and paste it in an appropriate subfolder within the Custom folder.
   b. Click the Edit link for the copied report.
3. Click Edit for the layout template to insert Quick Parts content into, and save the .rtf file to your computer with a new file name.
4. Open the .rtf file with Microsoft Word.
5. Put your cursor where you want to insert new content.
6. From the Insert tab on the ribbon, click Quick Parts within the Text group, and select the component to insert.
7. Edit the inserted component as needed and add any other components.
8. Save your changes in Word.

Previewing the Layout Template Changes
To preview your edits before uploading your layout template to the BI catalog:
1. On the ribbon, open the BI Publisher tab and click Sample XML within the Load Data group to import sample data from the data model. Skip this step if you already loaded sample data.
2. At the beginning of the document, replace the path with the location of the downloaded subtemplate file on your computer. For example, change "<?import:xdoxsl:///Common Content/Templates/Workflow Notification Subtemplate.xsb?>" to "<?import:file:///C:/Template_Directory/FinFunWorkflowNotificationSub.rtf?>".
3. From the BI Publisher tab on the ribbon, click HTML in the Preview group.
4. If the preview reflects your changes as expected, then change the path back to the original location.
5. Save your changes in Word.

Uploading the Modified Layout Template
To upload your layout template to the BI catalog after previewing the changes:
1. Back in the BI catalog, click Edit for the report within the Custom folder, if that page isn’t still open.
2. Click the View a list link.
3. Click the Create icon on the table toolbar.
4. In the Upload or Generate Layout section, click Upload.
5. Upload your edited .rtf file with a unique layout name.
6. Back on the page for editing the report, click Delete for the layout template that you downloaded earlier.
7. Click the Save Report icon.

Related Topics
- Configurable Email Notifications: Implementation Considerations
- How You Modify Copies of Predefined Reports
- Generate Sample Report Data
- Style Templates

Set Up Content to Appear in Only Email or In-App Workflow Notifications
For workflow tasks that have configurable email and in-app notifications, the same .rtf report layout template is used for both types of notifications. When you edit a copy of predefined templates in Microsoft Word to modify the notifications, you can make content conditional. For example, you can add an attribute from the data models used for the report, and set the attribute to appear only in in-app notifications.

The logo, action buttons, and links at the end of email notifications are predefined to appear only in emails, based on the subtemplate. The approval history is usually predefined to also appear in the body of only email notifications. Any conditional setting you apply to these components in the .rtf template won’t override the predefined setup.
Prerequisites
Generate sample report data from the data model used for the report, and save the .xml file to your computer.

Defining Conditional Regions
To define a conditional region of content that appears only in email or in-app notifications:

1. Open your .rtf report layout template in Microsoft Word.
2. On the ribbon, open the BI Publisher tab and click Sample XML within the Load Data group.
3. Select the .xml file you downloaded to import sample data from the data model.
4. In your .rtf document, select the content you want to make conditional.
5. On the ribbon, click Conditional Region within the Insert group.
6. In the Conditional Region dialog box, on the Properties tab, select BINDISONLINENOTIF from the Data field list in the General section. The values in this list come from the sample data you imported from the data model.
7. Select Date/Text from the next list.
8. In the Condition 1 section, select Equal to from the Data field list.
9. In the corresponding text field, enter true for content to appear only in in-app notifications, or false for content to appear only in emails.
10. Make sure that form fields containing the conditional logic are inserted around your selected content. The beginning form field, c, should be immediately before the conditional content, and the closing form field, ec, should be immediately after. Move the form fields as needed.

   Tip: To make sure you're looking at the correct form fields, double-click the c form field to open the Conditional Region dialog box and see the BINDISONLINENOTIF setting.

11. Save your changes in Word.

Entering Conditional Code
If the data model for your report doesn't have the BINDISONLINENOTIF attribute, then:

1. In your .rtf report layout template, put your cursor immediately before the content you want to make conditional.
2. Enter the following code, which functions the same as the c form field:
   - `<if:BINDISONLINENOTIF='true'?>` for in-app only
   - `<if:BINDISONLINENOTIF='false'?>` for email only
3. Put your cursor immediately after your conditional content.
4. Enter `</if:?>`, which functions the same as the ec form field.
5. Save your changes in Word.

Related Topics
- Generate Sample Report Data

Preview Changes to Layout Templates for Workflow Notifications
To modify workflow notifications, you edit a local copy of the .rtf report layout templates in Microsoft Word. Before uploading the .rtf files to the business intelligence (BI) catalog, you should preview the output with the changes you made. You can avoid uploading a broken report that displays an error in the notifications sent to users.

   Note: The exact steps can vary depending on your version of Microsoft Word.
Prerequisites

- Generate sample report data from the data model used for the report, and save the .xml file to your computer.
- Download a local copy of the subtemplate that applies to your own report layout template:
  a. In the BI catalog, expand Shared Folders > Custom > Common Content > Templates if you're using a modified subtemplate, or Shared Folders > Common Content > Templates for the predefined subtemplate.
  b. Click Edit for Workflow Notification Subtemplate.
  c. In the Templates section, click the link in the Locale column.
  d. Save the subtemplate .rtf file to your computer.

Previewing Output

To generate sample output from a local layout template:

1. Open your .rtf report layout template in Microsoft Word and make your edits.
2. On the ribbon, open the BI Publisher tab and click Sample XML within the Load Data group.
3. Select the .xml file you downloaded to import sample data from the data model.
4. At the beginning of your .rtf document, replace the path with the location of the downloaded subtemplate file on your computer. For example, change `<?import:xdoxsl:///Common Content/Templates/Workflow Notification Subtemplate.xsb?>` to `<?import:file:///C:/Template_Directory/FinFunWorkflowNotificationSub.rtf?>`.
5. From the BI Publisher tab on the ribbon, click HTML in the Preview group.
6. If the preview reflects your changes as expected, then change the path back to the original location.
7. From the BI Publisher tab on the ribbon, click Validate Template in the Tools group.
8. Also in the Tools group, click Check Accessibility.
9. Save your changes in Word.

Related Topics

- Generate Sample Report Data

Manage Workflow Rules Using a Spreadsheet

Create Workflow Rules Using a Spreadsheet

The Simplified Workflow Rules Configuration feature is a spreadsheet based alternative to creating rules in Oracle Business Process Management (BPM). You can use spreadsheet templates available on the Manage Workflow Rules in Spreadsheet page to manage rules for Financials application workflows.

Note: Currently, only Payables Invoice Approval and General Ledger Journal Approval workflows use this feature.
To create workflow rules in a spreadsheet, perform the following steps:

1. Sign in and navigate to the Manage Workflow Rules in Spreadsheet task.
2. Download the rule template from the Rule Templates section of the Manage Workflow Rules in Spreadsheet page.
3. Define the workflow rules in the spreadsheet.
4. Generate the rule file.
5. Upload the rule file to create rules.
6. Verify the spreadsheet upload.

**Caution:** You must use MS Excel version 2016 to create workflow rules. Also, every successful rule upload using a spreadsheet template overrides the existing rules for the workflow.

**Tip:** Once you create rules using the rule templates, we recommend you use the spreadsheet method only for any future maintenance of rules.

### Manage Workflow Rules Using a Spreadsheet

Before creating and managing workflow rules, perform the following steps:

1. Sign in to the application as a Financial Application Administrator.
2. Verify if the Approval Routing Administration feature is enabled. If it isn't enabled, navigate to the Offerings work area and enable the Approval Routing Administration feature as follows:
   - Offering: Financials
   - Functional Area: Financials
   - Feature: Approval Routing Administration
3. In the Setup and Maintenance work area, use the following:
Download the Rules Template

To download the rule template, perform the following steps:

1. In the Rule Templates section of the Manage Workflow Rules in Spreadsheet page, select the required workflow.
2. Click Download. The Download Templates dialog box appears.
3. From the dialog box, select the required template. Save the template to your local computer.

⚠️ **Caution:** You must use MS Excel version 2016 to create workflow rules.

🎬 **Note:** Each rule template contains an example of an approval business case to demonstrate how to manage workflow rules using the rule template.

Define the Rules in the Spreadsheet

After downloading the rule template, you must define the workflow rules using the sheets provided in the rule template. The rule template spreadsheet has the following sheets:

1. **Instructions**: This sheet contains details of the help topics present on Oracle Help Center for this feature and the Generate Rule File button. You can also update your rule template version from the Instructions sheet.
2. **Workflow Rules**: Provides a template for configuring transaction approval rules.

🎬 **Note:** The name of this sheet varies for each product. For example, for Payables, the sheet is labeled as Invoice Approval Rules.

3. **Data Set**: This sheet provides a template to map the varying attributes to the data.

🎬 **Note:** Currently, data sets are only available for Payables workflows. For more information on data sets, refer to the Data Sets section.

A business rule is an approval requirement within your approval policy. Before defining rules in the rule template, you must analyze approval policy. Consider the following points before defining a business rule:

- Which transactions require approval?
- Who approves transactions in your organization?
- Do the approvers vary based on the transaction attributes? If so, use a data set.
- What are approval conditions?
- How do you want to route the approval notifications?
- Which approvals require FYI notifications?
- Which transactions are exempted from the approval rule?

For example, for Payables, if your organization's approval policy mandates that:

- All invoices that aren't matched to a purchase order must be approved by two levels of the supervisory hierarchy starting from the manager of the user who creates the invoice.
All invoices that have an invoice amount of more than 5000 USD must be approved by a group of personnel from the Finance department.

For this example, you need two business rules. To define approval rules, use the Workflow Rules sheet. Enter the details in the following sections:

1. **Rule Description**

   Enter the description for each approval business rule that you define.

2. **Approvers**

   In this section, designate approvers and specify approval routing. The template supports a variety of approval routing options. The following table provides details on approval routing and how it works.

<table>
<thead>
<tr>
<th>Approval Routing</th>
<th>How Approval Routing Works</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervisory Hierarchy</td>
<td>Members of the supervisory hierarchy beginning from the first applicable approver receive approval notifications.</td>
</tr>
<tr>
<td>Group in Parallel</td>
<td>Members of an approval group receive approval notifications. All members receive notifications at the same time. All members must take an action on the approval notification.</td>
</tr>
<tr>
<td>Group in Serial</td>
<td>Members of an approval group receive approval notifications. Only when a member takes an action on the approval notification does the next member of the series receive the approval notification.</td>
</tr>
<tr>
<td>Group First Responder</td>
<td>Members of an approval group receive approval notifications. All members receive notifications at the same time. Only one member needs to take action on the approval notification.</td>
</tr>
<tr>
<td>Job Level Hierarchy</td>
<td>Members of the job hierarchy beginning from the first applicable approver receive approval notifications.</td>
</tr>
<tr>
<td>User</td>
<td>The specified application user receives the approval notification.</td>
</tr>
<tr>
<td>Role</td>
<td>The users with the specified application role receive the approval notification.</td>
</tr>
<tr>
<td>Auto Approve</td>
<td>Transactions that are automatically approved. No notifications are sent.</td>
</tr>
<tr>
<td>Auto Reject</td>
<td>Transactions that are automatically rejected. No notifications are sent.</td>
</tr>
<tr>
<td>FYI</td>
<td>Information only notifications. No action is required from the approver.</td>
</tr>
<tr>
<td>Skip Approval</td>
<td>Transactions for which the rule isn't applicable. No notifications are sent.</td>
</tr>
</tbody>
</table>
Note: You can only use an approval group that exists in the BPM.

For detailed instructions on the other columns in the Approvers section, refer to the tool tip on each column header.

3. Approval Conditions

In the Approval Conditions section you can select the attributes based on which the transaction should be evaluated for the workflow rules. You can also add attribute categories.

To add an attribute category:

a. Open the list of values associated with the last column in the Approval Conditions section.

b. Select the required attribute category.

To add an attribute:

a. Open the list of values associated with the attribute category.

b. Select the required attribute.

For example, for Payables, in the Invoice Approval template, you can select attributes such as Business Unit and Invoice Amount from the list of values associated with attribute category Invoice Header. Similarly, you can select attributes for categories, such as Invoice Line, Invoice Distributions, and more.

While defining approval conditions, you can use a variety of operators. The following table lists the supported operators.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Value Type</th>
<th>Format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute is a specific value</td>
<td>Text, number, or date</td>
<td>value</td>
<td>If the Invoice Type is Standard, then enter the value as: Standard</td>
</tr>
<tr>
<td>Attribute value is one of multiple specific values</td>
<td>Text or number</td>
<td>in (value 1, value 2, ...)</td>
<td>If the BU name is Vision Operations, Vision Services, or Vision Foods then enter the value as: Vision Operations, Vision Services, Vision Foods</td>
</tr>
<tr>
<td>Attribute value should be within a range of values</td>
<td>Number or date</td>
<td>between value 1 and value 2 OR value 1 to value 2</td>
<td>If the Invoice Date is between 01 August 2018 to 01 August 2019, then enter the value as: Between 01/08/2018 and 31/08/2019 OR 01/08/2018 to 31/08/2018</td>
</tr>
<tr>
<td>Condition</td>
<td>Value Type</td>
<td>Format</td>
<td>Example</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>------------</td>
<td>------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Attribute value starts with a specific value</td>
<td>Text</td>
<td>Starts with value</td>
<td>If the BU name starts with Vision then enter the value as:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Starts with Vision</td>
</tr>
<tr>
<td>Attribute value ends with a specific value</td>
<td>Text</td>
<td>Ends with value</td>
<td>If the BU name ends with Operations then enter the value as:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Ends with Operations</td>
</tr>
<tr>
<td>Attribute value contains a specific value</td>
<td>Text</td>
<td>Contains value</td>
<td>If the Pay Group contains Standard then enter the value as:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Contains Standard</td>
</tr>
<tr>
<td>Attribute value matches a specific value</td>
<td>Text</td>
<td>Matches value</td>
<td>If the Description matches manual invoice then enter the value as:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Matches manual\s(\s.*,\s) invoice</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>In this example, the Matches operator begins with Manual and ends with Invoice. Between the two words, there can be one space and any character.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Other options that can be used with the Matches operator are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(.*) - Denotes zero or more characters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(.+) - Denotes one or more characters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>\s - Denotes space.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>\d - Denotes numbers from 0-9.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>? - Makes a character optional. For example: \d?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[ ] - Specifies range such as A-Z, 0-9</td>
</tr>
<tr>
<td>Attribute value is more than or equal to a</td>
<td>Number</td>
<td>More than equal</td>
<td>If the Invoice amount is more than or equal to 500, then enter the value as:</td>
</tr>
<tr>
<td>specific number</td>
<td></td>
<td>to number OR</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Define Approval Management

<table>
<thead>
<tr>
<th>Condition</th>
<th>Value Type</th>
<th>Format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute value is less than or equal to a specific number</td>
<td>Number</td>
<td>Less than equal to number OR &lt;= number</td>
<td>If the Invoice amount is less than or equal to 500, then enter the value as: Less than equal to 500 OR &lt;= 500</td>
</tr>
<tr>
<td>Attribute value is more than a specific number</td>
<td>Number</td>
<td>More than number OR &gt;number</td>
<td>If the Invoice amount is more than 500, then enter the value as: More than 500 OR &gt;500</td>
</tr>
<tr>
<td>Attribute value is less than a specific number</td>
<td>Number</td>
<td>Less than number OR &lt;number</td>
<td>If the Invoice amount is less than 500, then enter the value as: Less than 500 OR &lt;500</td>
</tr>
<tr>
<td>Attribute value is on or before a specific date</td>
<td>Date</td>
<td>On or before date</td>
<td>If the Invoice date is on or before 01/10/2018, then enter the value as: On or before 01/10/2018</td>
</tr>
<tr>
<td>Attribute value is on or after a specific date</td>
<td>Date</td>
<td>On or after date</td>
<td>If the Invoice date is on or after 01/10/2018, then enter the value as: On or after 01/10/2018</td>
</tr>
<tr>
<td>Attribute value is before a specific date</td>
<td>Date</td>
<td>Before date</td>
<td>If the Invoice date is before 01/10/2018, then enter the value as: Before 01/10/2018</td>
</tr>
<tr>
<td>Condition</td>
<td>Value Type</td>
<td>Format</td>
<td>Example</td>
</tr>
<tr>
<td>--------------------------------------------------------------------------</td>
<td>--------------</td>
<td>-----------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Attribute value is after a specific date</td>
<td>Date</td>
<td>After date</td>
<td>If the Invoice date is after 01/10/2018, then enter the value as: After 01/10/2018</td>
</tr>
<tr>
<td>Attribute value is a specific value and the condition must be evaluated as case insensitive</td>
<td>Text</td>
<td>Equals ignore case value</td>
<td>If the Invoice Source is equal to Manual, then enter the value as: Equals ignore case manual</td>
</tr>
</tbody>
</table>

**Note:** The operators aren't case sensitive. However, you must enter the date in the DD/MM/YYYY format only.

If you have a negative approval condition, add Not as a prefix to any of the supported operators. For example, your approval condition states that BU name isn’t Vision Operations, Vision Services, or Vision Foods then enter the value as: Not In (Vision Operations, Vision Services, Vision Foods).

If you have two distinct rule conditions that require the same approval routing, then you must enter the rule conditions in two separate rows. Ensure that the information in the Approvers section is identical for both the rows.

**Rule Blocks**

A rule block is a group of rows in the workflow rules spreadsheet where you define a business rule and all aspects of the business rule. Use a separate block for each business rule.

While all rule aspects defined within a rule block are processed simultaneously, rule blocks are processed in sequence. Therefore, before defining the rules, you must consider the sequence in which the rules should be processed.

You can create additional rows in a block and additional blocks in a sheet as needed.

To insert more blocks in a rule block:

1. Select a row.
2. Right-click and select **Add Block**.

To add a rule block after the existing rule blocks, click **Add Block** in the sheet.

To insert more rows in a rule block:

1. Select a row and right-click.
2. From the menu, select **Insert**.

To delete a rule block:

1. Select all the rows in a rule block.
2. Right-click and select **Delete Block**.

**Data Sets**

In your approval policy, if the approver of a transaction varies based on the transaction attributes, then you should use a data set. A data set lets you define a mapping between your data and the variation in approvers based on such data.
For example, a transaction with an amount greater than 5000 USD needs to be approved by an approval group. However, the approval group varies depending on the cost center. In this case, you can use a data set to define a mapping between the cost center and the approval group.

![Note: Currently, data sets are only available for templates for Payables Invoice Approval Workflow.]

To define a data set, perform the following steps:

1. Open the Data Set sheet of the rule template.
2. In the Set Name column, enter a unique name.
3. Depending on the approval routing of the rule for which you're using the data set, enter the value in the Approval Group/Supervisory Level/Job Level Range/User/Role column.

   For the given example, specify the approval group name in the Approval Group/Supervisory Level/Job Level Range/User/Role column.

![Note: You can only use an approval group that already exists in the BPM.]

4. In the Varying Attribute section, select the attributes based on which the approver varies for the transaction.

   For the given example, select Distribution Cost Center Segment from the list of values and specify the cost center values for each approval group.
5. Enter values for each varying attribute. You can also use the supported operators with the values.
6. Click Add New Column to create additional columns for varying attributes.
7. Click Add Data Set to create additional data sets.

After you create a data set, you must enter a data set reference in your rule in the Workflow Rules sheet by prefixing the data set name with $. For example, if you want to reference a data set named Supervisory, then in the workflow Rules sheet, enter the value as $Supervisory Set.

You can enter the data set references in the Approvers section of the Workflow Rules sheet. Based on the approval routing used for the rule, you can enter data set references in the following columns:

- Job Level Range
- Approval Level
- Group/User/Role Name

**Generate Rule File**

After entering the data in the Workflow Rules sheet, click the Generate Rule File button located in the Instructions sheet to generate the rule file. A compressed file is generated. Save the file in your local computer.

**Upload the Rule File**

To upload the rule file, perform the following steps:

1. Navigate to the Manage Workflow Rules in Spreadsheet page.
2. In the Rule Templates section, select the required workflow.
3. Click Upload. The Upload File dialog box appears.
4. In the File field, click Choose File.
5. From your local directory, choose the compressed rule file that was generated from the workflow rules template.
6. Click Submit. A confirmation message stating the process ID appears.
**Caution:** Every successful rule upload using a spreadsheet template overrides the existing rules for the workflow.

7. Click **OK**.
8. Check the status of the upload in the Upload History section.

### Update the Rule Template Version

While uploading your rule template, if you're asked to update the file version, perform the following steps:

1. Download the latest version of the rule template from the Manage Workflow Rules in Spreadsheet page. You can select any of the available templates for the workflow.
2. In the Instructions sheet of the rule template, click **Update Spreadsheet**.
3. Select the older version of the rule template and click **OK**.
   - This copies rules from the older version of your rule template to the latest version.
4. Review the copied rules and proceed as usual to create rules using the latest version of the rule template.

### Verify the Spreadsheet Upload

The Upload History section displays details of the spreadsheet uploads such as the date, user, rule template used, and the status.

If the rule upload process fails, the status is displayed as **Error**. Click **Error** to download the **Error CSV file**. Review the error details, resolve the errors in the spreadsheet, and generate the rule file again.

**Related Topics**
- Configure Offerings
- Update Existing Setup Data

### Modify Workflow Rules Using a Spreadsheet

After creating the rules using a spreadsheet template, you can modify them using a spreadsheet.

**Note:** Each time you make modifications, a new set of rules are created. The new rules overwrite the existing rules.

To modify workflow rules using a spreadsheet, perform the following steps:

1. Navigate to the Manage Rules in Spreadsheet page.
2. In the Rule templates section, select the required workflow.
3. For the workflow, select the link in the Last Successful Upload column. Save the copy of the last successfully uploaded rule template to your local computer.
4. Make the necessary changes in the spreadsheet and click **Generate Rule File**. A compressed file is generated. Save the generated rule file in your local directory.
5. On the Manage Rules in Spreadsheet page, select the required workflow in the Rule Templates section.
6. Click **Upload**. The Upload File dialog box appears.
7. In the **File** field, click **Choose File**.
8. From your local directory, select the compressed rule file to be uploaded for the rule creation that was generated from the workflow rules template.
9. Click **Submit**. A confirmation message stating the process ID appears.
10. Click **OK**.
11. Check the status of the upload in the Upload History section.
Note: If the upload fails, the status is displayed as Error. Click Error to download the Error CSV file. Review the error details, resolve the errors, and generate the rule file again.

FAQs for Manage Workflow Rules Using a Spreadsheet

What version of MS Excel is supported for rule creation?
Currently, only MS Excel version 2016 is supported for rule creation.

How can I generate the rule file??
After entering the data in the Workflow Rules sheet, click the Generate Rule File button in the Instructions sheet to generate the rule file. A compressed file is generated. Save the file in your local computer.

How can I make my existing BPM rules available in the rule template?
You can't move your existing BPM rules to a rule template automatically. You must enter them in the rule template manually and upload it to create rules.

Why can't I see the rules that I created using the rule template in BPM?
Here are two reasons why you can't see the rules:
- Your rule file upload failed. Check its status on the Manage Workflow Rules in Spreadsheet page.
- You are viewing workflow rules in edit mode in BPM. On the Task Configuration tab, click Reset to see the updated rules.

What happens to my existing rules when I successfully upload a rule file for a workflow?
With each successful upload, all existing rules for that workflow are overwritten. You must download the last successfully uploaded rule file and use it to make any modifications when required.

Change Workflow Task Titles

Users see titles for workflow tasks in many places, such as the notifications list in the global header, or the Worklist: Notifications and Approvals work area. You can change task titles and even set up titles for specific languages. So for example, people using the application in Spanish see the task title you define for Spanish.
1. In the Setup and Maintenance work area, go to the Manage Task Configurations task or another approval setup task in the Application Extensions functional area or another functional area.
2. In BPM Worklist, on the Task Configuration tab, select the workflow task.
3. Click the Edit task icon in the Tasks to be configured toolbar.

**Tip:** At any time before you click the Commit task icon in the Tasks to be configured toolbar, you can still click the Reset icon to go back to the predefined title.

### Define a Single Title for All Languages

If you want different titles for specific languages, skip this section and follow the steps in the next section.

1. On the General subtab, click the first Title icon after the Title field.
2. Use the expression builder to define what's shown as the title at runtime.
3. Click OK.
4. Click the Commit task icon in the Tasks to be configured toolbar when you're ready to roll out your changes.

### Define Different Titles for Specific Languages

Repeat these steps for each language where you want to override the predefined title.

1. On the General subtab, click the second Title icon after the Title field.
2. In the Translations dialog box, select a language in the Locale list.
3. Click the Add Row icon.
4. In the Key column, enter whatever you want to identify this title with. If you choose to use this title, then the key is shown in the Title field on the General subtab. Users won't see it.
5. In the column with the locale name, enter the title that users see. If you want the title to have variables, then use numbers in braces for each variable, starting with 0. For example, your title can look something like this: Request from {0} for Transaction {1}.
   a. Click the Edit icon if you have any variables.
   b. In the Edit Arguments dialog box, click the Expression Builder icon in the Value column.
   c. Use the expression builder to define what's shown in the title at runtime instead of the variable.
   d. Click OK to get back to the Translations dialog box.
6. Optionally add more rows and define more titles in the Translations dialog box for this language, for example, if you have other titles you might want to use later.
7. In the Translations dialog box, select the title to use for this language, and click OK.
8. Click the Commit task icon in the Tasks to be configured toolbar when you're ready to roll out your changes.
Workflow Task Life Cycle Setup

Define When to Automatically Dismiss or Withdraw Workflow Tasks

Only workflow tasks with a final status, such as Completed or Withdrawn, can be purged and removed from users' worklists. Tasks go from the Assigned status to the Completed status when the final assignee approves or rejects the tasks, or, with for your information (FYI) tasks, when assignees explicitly dismiss the tasks. If assignees don’t take actions that result in a final task status, within a certain period of time, then the tasks are automatically dismissed (FYI tasks) or withdrawn (all other tasks).

When Tasks are Eligible for Automatic Dismissal or Withdrawal

The FYI Notifications Expiration Period profile option determines when FYI tasks are eligible for automatic dismissal. In the Setup and Maintenance work area, use the following:

- Functional Area: Application Extensions
- Task: Manage Applications Core Administrator Profile Values

On the Manage Applications Core Administrator Profile Values page:

- Leave the profile option with the default value of 7, or replace it with a different number.
- The profile value represents the number of days after the FYI task is created.

When assignees don’t read or dismiss an FYI task within the specified number of days after the task was created, the task is then eligible to be automatically dismissed. All other tasks are eligible for automatic withdrawal when assignees don't take action to send the task to a final status within six months after the task was created.

When Eligible Tasks Are Automatically Dismissed or Withdrawn

Different processes run to automatically dismiss eligible FYI tasks or withdraw all other eligible tasks.

- **FYI Tasks**: The process runs every three days, starting the first day of each month. For example, it runs on May 1, 4, 7, and so on, and again on June 1 and every three days after. So, if you leave the FYI Notifications Expiration Period profile value at 7, then depending on when the process runs, an FYI task can be automatically dismissed within seven to ten days after it’s created. The process changes the FYI task status from Assigned to Completed.

- **All Other Tasks**: The process runs every three days, starting the second day of each month. For example, it runs on May 2, 5, 8, and so on, and again on June 2 and every three days after. The process changes the status of eligible tasks to Withdrawn.

Related Topics

- Set Profile Option Values
- Update Existing Setup Data
How Workflow Tasks Are Archived and Purged

Workflow tasks with a final status, such as Completed or Expired, can be archived and purged. Archiving keeps a copy of the task data for audit, data retention or analysis, and other purposes. Purging removes the completed tasks from users’ worklists and permanently deletes the original data.

Archive

Tasks are automatically archived once a month without you doing any setup. You can’t change or stop this automatic archive. You can, however, also run the Archive Workflow Tasks scheduled process as needed; for example, you need the latest data archived immediately for reporting purposes. The process includes all eligible tasks that aren’t yet archived.

Archived data includes task details, approval history, comments, and attachments. How you view or use the archived data depends on the products you’re using. For example, the data might be displayed in a table on a page, or available through a business intelligence subject area that you can select to create an analysis.

Purge

Archived tasks that were last updated over 30 days ago are immediately purged after the monthly automatic archive, without you doing any setup. You can’t change or stop this automatic purge.

Related Topics

- Submit Scheduled Processes and Process Sets

Set Up the Worklist Region on My Dashboard

You can add the Worklist: Notifications and Approvals region to My Dashboard, which is a blank dashboard by default. This region displays the workflow tasks assigned to the person using My Dashboard. After you add the Worklist region, select a value for the Welcome Dashboard Worklist Timeout Interval (ATK_HOME_PAGE_WORKLIST_TIMEOUT) profile option.

Adding the Region

To add the Worklist: Notifications and Approvals region to My Dashboard:

1. Click Navigator > My Dashboard.
2. Click your user image or name in the global header, and select Edit Pages in the Administration menu group.
3. Click the Add Content button where you want to place the region.
4. Open the Application Content folder in the Add Content dialog box.
5. Click Add for the Worklist: Notifications and Approvals item.
6. Click Close.
7. Save your work, and then click the Close button.
Defining the Timeout Interval

When users open My Dashboard and it contains the Worklist: Notifications and Approvals region, data for the region is retrieved. The Welcome Dashboard Worklist Timeout Interval profile option determines how long to continue retrieving before timing out and displaying no data. In the Setup and Maintenance work area, use the following:

- Functional Area: Application Extensions
- Task: Manage Application Toolkit Administrator Profile Values

**Note:** If you don't see this task, then make sure that the Application Toolkit Component Maintenance feature is enabled at the offering level in the Offerings work area.

On the Manage Application Toolkit Administrator Profile Values page, set the Welcome Dashboard Worklist Timeout Interval profile option.

- If you don't set a value for this profile option, which is blank by default, then the region doesn't time out.
- Retrieving data for the Worklist region affects the performance of My Dashboard as a whole. So, select a value for this profile option if your users have the Worklist region on My Dashboard and notice performance issues.

After the timeout, users can refresh the region to try retrieving the data again.

**Related Topics**

- Set Profile Option Values
- Configure Offerings
- Update Existing Setup Data

Best Practices for Approval Management

Use Direct Attribute Reference

During approval rules configuration, you may need to specify some transactional attributes. You can do this in two ways:

- Use a direct attribute reference.
- Use the attribute from the task payload object.

For example, consider the attribute Invoice Amount, which is a part of the Invoice Header object. Here’s how you can specify the attribute in the approval rule.

<table>
<thead>
<tr>
<th>Using Direct Attribute Reference</th>
<th>Accessing Attribute from Task Payload Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoice Header.Invoice Amount</td>
<td>Task.payload.getInvoiceHeader1Response.result.Invoice Amount</td>
</tr>
</tbody>
</table>

The recommended best practice is to always use a direct attribute reference when creating approval rules. This optimizes the rule evaluation process at runtime.
Create Rules in Advanced Mode

Financial transactional models consist of an object hierarchy where one parent object can have one or more child objects and each of these child objects can have one or more grandchild objects.

For example, the object hierarchy for Payables invoices consists of invoice header, lines, and distributions. If an attribute from a non-header level is used to configure rules, then the application evaluates each line or distribution object.

This works well if the business requirement is to evaluate the approval rule conditions for each line or distribution. However, if the business requirement is to evaluate if the rule condition holds true for any one line or distribution, then rules should be defined in advanced mode.

Advanced mode offers additional pattern-matching options. These options can optimize rule execution even when rule conditions include attributes from different object at different levels of object hierarchy. These are the pattern-matching options in advanced mode:

Advanced mode offers additional pattern-matching options. These options can optimize rule execution even when rule conditions include attributes from objects at different levels of object hierarchy.

Advanced Mode Pattern-Matching Options

Here are the pattern-matching options in advanced mode:

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
</table>
| Each         | - Evaluates each line or distribution to check if there is a match with the rule conditions.  
               | - Sends a notification for all the lines or distributions that satisfy the rule conditions. |
| At least one | - Evaluates the lines or distributions against the rule conditions until at least one match is found. Once a match is found, no further lines or distributions will be evaluated for the rule.  
               | - Sends a notification once the first match is found. Doesn't send a notification if no match is found. |
| None         | - Evaluates the lines and distributions to check that none of them match with the rule conditions.  
               | - Sends a notification only if there's no match with all the lines and distributions. |

Note: Do not use this with NOT operator.

| Aggregate    | - Allows use of a combination of operations, such as Sum, Count, Maximum and so on.  
               | - Use it when attributes from multiple attribute sources need to be checked for a single match. |
Use Null Checks for Optional Attributes

Among the transaction attributes used in rule conditions, there might be some attributes which are optional during transaction entry.

When creating rules using such optional attributes, it's recommended to include a null check condition. This ensures that rule evaluation doesn't fail for transactions which have no value entered for such optional attributes.

For example, for an approval rule condition using Purchase Order Number attribute for Payables Invoices:

```
InvoiceHeader.Purchase Order Number startsWith ABC
```

Add a null check condition to cover those invoices which don't have a purchase order number referenced on them.

```
InvoiceHeader.Purchase Order Number != null
```

It happens quite often that the user doesn't enter any value for an optional attribute on the UI. If an optional attribute is used in the approval rules, it will lead the server to throw a null pointer exception. The application will reject the transaction when such a rule is executed. It is highly recommended to include a null check operation wherever optional attributes are used.

Optimize Rule Evaluation

Business rules involve complex interactions between attributes from various objects. These interactions are sometimes repetitive in nature.

Cartesian product condition occurs when such a rule leads to the creation of many objects (or child facts). When a rule is configured using child and grandchild attributes, you need to establish the relationship between them to ensure optimum rule evaluation. When a rule is configured using both line and distribution attributes, then you need to establish the relationship between parent and child attributes to ensure that only the distributions for those lines that are meeting the line condition are evaluated. If such relationship isn't established, then all distributions for the invoice as a whole are evaluated for every line matching the rule conditions.

Workflow Transaction Console

Overview of Workflow Transaction Console for Financials

Use the workflow Transaction Console to monitor and troubleshoot workflow tasks for the Invoice, Expenses, and Journal Approval workflows.

From the console you can:

- View the latest status of all workflow tasks.
- Review the issue description and resolution for failed tasks.
- Take appropriate action on a failed task.
- Search tasks based on user-defined criteria.
- Download search results to a spreadsheet.
Related Topics

- Manage Workflow Transactions
- Statues for Filtering Transactions
- Actions for Managing Transactions

Give Users Access to Manage Financials Workflow Transactions

Users can manage transactions for the Invoice, Expenses, and Journal Approvals workflows from the Transaction Console.

Give Financial Users Administrator access

You have a couple of options for giving users access to the Transaction Console work area, depending on whether you’re assigning them predefined job roles, or your own configured job roles.

- Assign the predefined Financials Application Administrator (ORA_FUN_FINANCIAL_APPLICATION_ADMINISTRATOR_JOB) job role.
- Your own configured job role must include the Financial Transaction Approval Reviewing (ORA_FIN_REVIEW_APPROVAL_TRANSACTIONS) duty role.

Limit What Users Can See

By default, users who have access to the Transaction Console work area can see all transactions, from all product families.

To make sure that financial users view only the financial workflows, enable transaction security:

1. In the Setup and Maintenance work area, go to the Manage Enterprise HCM Information task.
   - Offering: Financials
   - Functional Area: Enterprise Profile
   - Task: Manage Enterprise HCM Information
2. On the Edit Enterprise page, click Edit and then select Update.
3. Complete the fields in the Update Enterprise dialog box and click OK.
5. Click Submit.

Note: You just need to do this for one offering. The setting now applies to all product families. If you’re also using Oracle HCM Cloud, make sure transaction security profiles are set up so that HCM administrators can see and act on HCM transactions.

Related Topics

- Assign Roles to an Existing User
- Edit Job and Abstract Roles
- Role Copying or Editing
7 Define Help Configuration

Set Up Help

You don’t have to set anything up for help to work. But you can do optional setup, for example to make help icons visible to everyone, or to give people access to add help content. First enable help features, then do some tasks in the Application Extensions functional area.

Enable Features

In the Offerings work area, review these features at the offering level.

- Local Installation of Help: Make sure to leave this enabled.
- Access to Internet-Based Help Features: This feature isn’t used, so it doesn’t matter if you enable it or not.
- Help Content Management: Enable this feature if you want some people to be able to add company-specific help to help windows or the Getting Started work area.
- Security for Added Help: Enable this feature if you want certain help to be available only to a restricted set of user roles.

Caution: Enable this feature only if you have this requirement, because the feature can affect performance.

Related Topics

- Configure Offerings
- How You Manage Different Types of Help

Show Help Icons by Default

Watch video

By default, help icons that open help windows are hidden. You can set it up so that users do see the icons every time they sign in, on any page that has help icons.

1. In the Setup and Maintenance work area, go to the Set Help Options task in the Application Extensions functional area.
2. On the Set Help Options page, select the Show help icons by default check box. You just need to do this for one offering, and the setting applies to everyone.
3. Sign out and sign back in to see the icons by default. The same goes for any user next time they sign in.

To hide the icons, people can still click their user image or name in the global header and select Hide Help Icons, and later select Show Help Icons to see the icons again.
Give People Access to Create and Edit Help

If you enabled the Help Content Management feature, check who has the Manage Help Content (ATK_CUSTOMIZE_HELP_TOPICS_PRIV) privilege. Here’s what these users can do:

- Add help in any help window, and also edit and manage the help that anyone added.
- Edit existing pages in the Getting Started work area and also add new pages.

By default, the administrators for product families have this privilege. Your security administrator can assign job roles with this privilege to other users. The Assign Help Text Administration Duty task is a reminder for you to follow up with your security administrator. Make sure that people who want to create and edit help have the access to do so.

Enable Global Search for Added Help

If your company adds help, you might want to enable global search and make sure it includes the Help category. People can then search for any added help using the search in the global area.

Create Groups to Limit Access to Added Help

In this example, you want to add company policies as help in a help window. But, at all times, only human resources (HR) specialists should be able to see those policies. Let’s look at the steps to create a help security group that includes only the job role for HR specialists. You can then secure your company policies with this help security group so that only people with that job role can see the help.

Before You Start

In the Offerings work area, make sure that these features are enabled at the offering level:

- Local Installation of Help
- Help Content Management
- Security for Added Help
Create the Help Security Group

1. In the Setup and Maintenance work area, go to the Manage Help Security Groups task in the Application Extensions functional area.
3. Enter these values, but leave the start and end dates blank.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help Security Group</td>
<td>HR</td>
</tr>
<tr>
<td>Display Name</td>
<td>HR Only</td>
</tr>
<tr>
<td>Description</td>
<td>Viewing by HR specialists only</td>
</tr>
</tbody>
</table>

4. Click Save.
5. With your new help security group selected, go to the Associated Roles section and add a new row.
6. Select PER_HUMAN RESOURCE SPECIALIST as the role name.
7. Click Save and Close.

Related Topics
- Configure Offerings
- Update Existing Setup Data
- Add Your Content to Help Windows
8 Define Application Toolkit Configuration

Overview of Application Toolkit Configuration

Oracle Fusion Application Toolkit (ATK) provides many components that are available to users of all product families. These components include Applications Help, the Reports and Analytics pane, and the Watchlist. In the Setup and Maintenance work area, use the Application Toolkit tasks in the Application Extensions functional area to set up some of these components.

Note: The tasks are available only if the Application Toolkit Component Maintenance feature is enabled.

Tasks

Use these tasks in the Application Extensions functional area:

- **Map Reports to Work Areas**: Determine what's available in the Reports and Analytics pane for specific work areas.
- **Set Watchlist Options**: Define settings that affect what's displayed in the Watchlist and how often items are refreshed.
- **Manage Application Toolkit Administrator Profile Values**: Set profile options to affect how some Application Toolkit components work.
- Use other Application Toolkit tasks in this functional area to set up help:
  - Set Help Options
  - Assign Help Text Administration Duty
  - Manage Help Security Groups

Related Topics

- Set Up the Mapping Service for Contextual Addresses
- Set Up the Worklist Region on My Dashboard
- Set Up Help

Map Reports to Work Areas

Set Up the Reports and Analytics Panel Tab

You can find the Reports and Analytics panel tab in many work areas, and the analytics and reports you see in it depend on the work area. You can define what's available for a specific work area, by mapping reports from the business intelligence (BI) catalog to that work area. In this mapping context, reports refer to both analytics and reports. Your changes apply to all users who have access to the work area you're mapping.
Mapping Reports from Your Work Area

To map reports to the work area that you're in:

1. Expand the Reports and Analytics panel tab.
2. Click the **Edit Settings** icon in the panel tab.
   
   You see all the reports that are currently mapped to your work area.
3. Click **Select and Add**.
4. Find the report in the catalog and select it.
5. Click **OK**.
6. To remove any mapping, select the report and click **Remove**.
7. Save your work.

Mapping Reports to Any Work Area

To map reports to any work area that you have access to:

1. In the Setup and Maintenance work area use the Map Reports to Work Areas task.
2. Select the application of the work area you want to map to.
3. Select the work area.
4. Click **Search** and see all the reports that are currently mapped to that work area.
5. Click **Select and Add**.
6. Find the report in the catalog and select it.
7. Click **OK**.
8. To remove any mapping, select the report and click **Remove**.

   **Tip:** Click **Synchronize** to remove all mappings to any reports that are no longer in the catalog. You synchronize all work areas, not just the one you’re mapping.
9. Save your work.

Set Up Reports for Scheduling

You can set up reports as scheduled processes, which means users can submit them from the Scheduled Processes and other work areas. If you want users to also submit these scheduled processes from the Reports and Analytics work area and panel tab, then you must configure properties for the corresponding reports.

Enabling a Report for Scheduling

To enable scheduling in the Reports and Analytics work area and panel tab:

1. In the Reports and Analytics work area or panel tab, edit the report in the **business intelligence catalog**.
2. Click **Properties**.
3. On the General tab in the Properties dialog box, enter the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise Scheduler Job Package Name</td>
<td>The path for the job definition, for example: / oracle/apps/ess/&lt;product family&gt;/ &lt;product&gt;/ &lt;business area&gt;/ Jobs</td>
</tr>
<tr>
<td>Enterprise Scheduler Job Definition Name</td>
<td>The job definition name (not display name), for example: ABCDEFG</td>
</tr>
</tbody>
</table>
FAQs for Map Reports to Work Areas

Why can't I see reports when mapping reports to work areas for the Reports and Analytics panel tab?
Either no reports are currently mapped to the work area you select on the Map Reports to Work Areas page, or you don't have access to the reports that are mapped.
Similarly, when you're selecting a report to map, you can see only the reports that you have access to. Ask your administrator to either:
- Assign you roles that have access to the reports you want to map to work areas.
- Grant the Reports and Analytics Region Administration Duty to someone who already has access to those reports.

Why can't I see reports when I edit settings for the Reports and Analytics panel tab?
In the Edit Settings window, you might not see a currently mapped report because you don't have access to it.
Similarly, when you're selecting a report to map, you can see only the reports that you have access to. Ask your administrator to either:
- Assign you roles that have access to the reports you want to map to work areas.
- Grant the Reports and Analytics Region Administration Duty to someone who already has access to those reports.

Set Watchlist Options

Disable or Enable Watchlist Categories and Items
To disable or enable predefined Watchlist categories and items for all users, use the following in the Setup and Maintenance work area:
- Functional Area: Application Extensions
- Task: Set Watchlist Options
Ultimately, what users see in their own Watchlist would be the categories and predefined items that you enable in the Set Watchlist Options page:
- Plus any saved searches that the user is using as Watchlist items
Any Category or Item
When you disable any category or item, you also disable the processes that calculate the Watchlist item counts. These processes include creating data caches, performing security checks, calling services across domains, running queries, and so on.

Predefined Watchlist Items
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 isn't available for users to display in their Watchlist.
- The item is removed from any Watchlist where it's currently displayed.

If you disable a Watchlist category, then the category isn't available for users to include in their Watchlist. All Watchlist items within the category are also disabled.

User-Created Saved Search Watchlist Items
A Watchlist item with the User-Created Saved Search type doesn't appear in the Watchlist. It controls the display of the Manage Watchlist button on pages with saved searches. If you disable this type of Watchlist item, then:
- The Manage Watchlist option isn't available on the corresponding page, so users can't use their own saved searches as Watchlist items.
- Any user-defined saved searches (from that page) already used as Watchlist items are removed from the users' Watchlist. The saved searches are still available for searching, but not for the Watchlist.

Watchlist Category
If you disable a Watchlist category, then:
- The category isn't available for users to include in their Watchlist.
- All Watchlist items within the category are also disabled.

Note: You can see the Watchlist icon in the global header only if your default home page layout is Panel or Banner.

Refresh Intervals for Watchlist Items
All Watchlist items have a predefined refresh interval, which controls how often the query that calculates the count for a Watchlist item can run. You can use the Set Watchlist Options page to edit the interval values. In the Setup and Maintenance work area, use the following:
- Functional Area: Application Extensions
Task: Set Watchlist Options

How the Refresh Works
The count for any Watchlist item gets refreshed as follows.

- When users open the Watchlist in the global header for the first time after signing in, the item is refreshed if five minutes have passed since its last refresh in an earlier session.
- When users open the Watchlist again, the item is refreshed only if the time since its last refresh is equal to or greater than the refresh interval.

User-Created Saved Search Watchlist Items
What you enter as the refresh interval for a Watchlist item of type User-Created Saved Search applies to all Watchlist items based on saved searches that users create for that item. For example, you set the refresh interval for the Corporate Card Transactions item to five minutes. Multiple users create saved searches for corporate card transactions and use those saved searches as Watchlist items in their own Watchlist. All of these Watchlist items would have a refresh interval of five minutes.

Related Topics
- Create Watchlist Items

FAQs for Set Watchlist Options

How can I change predefined Watchlist category and item names?
Edit the standard lookup type that stores the predefined Watchlist category and item names.

1. Click Navigator > Setup and Maintenance.
2. In the Setup and Maintenance work area, go to the following:
   - Functional Area: Application Extensions or a product-specific functional area
   - Task: Manage Standard Lookups
3. On the Manage Standard Lookups page, find the lookup type for the Watchlist category you want to edit. The lookup types for predefined categories end with WATCHLIST, for example EXM_EXPENSES_WATCHLIST.
4. Edit the lookup type meaning to change the category name.
5. To change item names, edit lookup code meanings.
6. Save your work.

Related Topics
- Overview of Lookups
9 Define Enterprise Scheduler Job Definitions and Job Sets

Overview of Managing Job Definitions and Job Sets

Users run scheduled processes based on Oracle Enterprise Scheduler Services to process data and, in some cases, to provide report output. Using the Enterprise Scheduler Jobs tasks, you can define job definitions, list of values sources, and job sets.

The various Enterprise Scheduler Jobs tasks are:

- Manage Enterprise Scheduler Job Definitions and Job Sets for Financial, Supply Chain Management, and Related Applications
- Manage Enterprise Scheduler Job Definitions and Job Sets for Human Capital Management and Related Applications
- Manage Enterprise Scheduler Job Definitions and Job Sets for Customer Relationship Management and Related Applications

Each task includes:

- **Job Definitions**: Contain the metadata about the job and the options available to users. A job definition is defined by a job type, such as Oracle Business Intelligence Publisher or PL/SQL Job type or others.
- **List of Values Sources**: Determine where a list of values for the parameters in a job definition comes from and what the specific values are.
- **Job Sets**: Collections of several jobs in a single process set that the users submit instead of running separate jobs.

To access the Manage Enterprise Scheduler Job Definitions and Job Sets tasks, use the following in the Setup and Maintenance work area:

- Functional Area: Application Extensions or a product-specific functional area
- Task: Manage Enterprise Scheduler Job Definitions and Job Sets

Related Topics

- Overview of Scheduled Processes

Manage List of Values Sources

A list of values source determines where a list of values comes from and what the specific values are. Use these lists to display values for the parameters and application defined properties in job definitions which the user can select. For example, a list of countries for a Country parameter. To manage list of values sources, open the Manage List of Values Sources tab using the Manage Enterprise Scheduler Job Definitions and Job Sets page. In the Setup and Maintenance work area, use the following:

- Functional Area: Application Extensions or a product-specific functional area
• Task: Manage Enterprise Scheduler Job Definitions and Job Sets

Editing and Deleting List of Values Source

You can edit and delete list of values sources using the Manage List of Values Sources tab. While the List of Values Source Definition Name is a fully qualified name of the view object, the User List of Values Source Name doesn't have validation. So, you can edit the **User List of Values Source Name** field and enter a name according to your preference.

**Note:** You can edit list of values sources for use only in job definitions that are not predefined.

Managing Job Definitions

Job Definitions

Scheduled processes are based on jobs that process data and, in some cases, provide output. Each job requires a **job definition**. A job definition can also include parameters and user properties that the user defines while submitting the scheduled process.

You can view, create, edit and duplicate job definitions on the Manage Job Definitions tab using the Manage Enterprise Scheduler Job Definitions and Job Sets page. In the Setup and Maintenance work area, use the following:

- Functional Area: Application Extensions a product-specific functional area
- Task: Manage Enterprise Scheduler Job Definitions and Job Sets

Viewing Job Definitions

Use the table on the Manage Job Definitions tab to view the job definitions created for the application. An asterisk in the name column indicates a predefined job definition.

This table describes the columns in the table on the Manage Job Definitions tab.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the job definition.</td>
</tr>
<tr>
<td>Display Name</td>
<td>Name of the job definition as available to users while submitting scheduled processes.</td>
</tr>
<tr>
<td>Description</td>
<td>Description for the job definition.</td>
</tr>
<tr>
<td>Path</td>
<td>The full directory path where the job definition is saved.</td>
</tr>
<tr>
<td>Execution Type</td>
<td>The type of job request for the given job definition, such as a Java, C, PL/SQL, Perl, or hosted script job.</td>
</tr>
<tr>
<td>Job Type</td>
<td>The name of the job type upon which the job definition is based.</td>
</tr>
</tbody>
</table>
Implementing Common Features for Financials and Project Portfolio Management

Chapter 9

Define Enterprise Scheduler Job Definitions and Job Sets

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Tip:** The table of job definitions shows only 10 to 20 items by default but you can use *Query by Example* to view other items.

**Parameters**

A parameter controls which records are included or how they are affected when the job runs. Parameters are available to users when they submit scheduled processes based on your job definitions. For example, a job updates only the records that are effective after the date that users enter in a Start Date parameter. You can create, edit and delete parameters for job definitions that are not predefined.

**User Property**

A user property is set in the job definition to attain some specific results.

**Related Topics**

- Overview of Scheduled Processes

**Create Job Definitions**

A *job definition* is defined by a Job type, such as Java or Oracle Business Intelligence (BI) Publisher reports. You can only create or duplicate job definitions which are based on BI Publisher reports, so that users can run the reports as scheduled processes. For more information about reports, see the Creating and Administering Analytics and Reports guides for your products.

**Creating Job Definitions**

Follow these steps:

1. Click **Navigator > Setup and Maintenance**.
2. In the Setup and Maintenance work area, go to the following:
   - Functional Area: Application Extensions or a product-specific functional area
   - Task: Manage Enterprise Scheduler Job Definitions and Job Sets
3. On the Manage Enterprise Scheduler Job Definitions and Job Sets page, open the Manage Job Definitions tab.
4. Click the **Create** icon.
5. In the Job Definition section, complete the fields, some of which are shown in this table.

<table>
<thead>
<tr>
<th>Field</th>
<th>What You Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Name</td>
<td>Provide a name that the users see while submitting the scheduled process.</td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name with only alphanumeric characters, for example, AtkEssPrograms1. A job definition name can't contain space or any special characters.</td>
</tr>
<tr>
<td>Job Application Name</td>
<td>Select the name of the application to associate the job definition with.</td>
</tr>
<tr>
<td>Field</td>
<td>What You Enter</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Job Type</td>
<td>Select BIPJobType only.</td>
</tr>
<tr>
<td>Report ID</td>
<td>Specify the path to the report in the catalog, starting with the folder within Shared Folders, for example: User-Defined/ &lt;Family Name&gt;/ &lt;Product Name&gt;/ &lt;Report File Name&gt;.xdo. Make sure to include the .xdo file extension for the report definition.</td>
</tr>
<tr>
<td>Default Output Format</td>
<td>Select the format of the output.</td>
</tr>
</tbody>
</table>

**Note:** Don’t select the **Enable submission from Enterprise Manager** check box.

6. Use the Parameters tab to define parameters as required.
7. The only user property you need to define is **EXT_PortletContainerWebModule** and this user property is automatically created.

**Caution:** You must not create or edit a user property unless you have the accurate information that is required to create or edit one.

8. Click **Save and Close**.

When you create a job definition, the **privilege** with the same name as the job definition is automatically created. For example, for a job definition named **AtkEssPrograms**, the privilege is named **RUN_ATKESSPROGRAMS**.

**Duplicating Job Definitions**

Users can also create job definitions by duplicating existing job definitions, including parameters. To duplicate a job definition:

1. On the Manage Job Definitions tab, select the job definition you want to duplicate.
2. Click **Duplicate** to get another row in the table with the duplicate job definition.
3. Select the duplicate job definition and click **Edit**.
4. Enter the name and path.
5. You can update the parameters as needed.
6. Click **Save and Close**.

When you duplicate a job definition, you automatically create a view object of the same name in the list of values sources.

**Note:** The attribute validations present on the attributes in the parameters view object are not copied over.

**Edit Job Definitions**

You can only edit certain fields in predefined job definitions, as some of the fields are read only. However, you can edit all aspects of **job definitions** that are not predefined.
Editing Job Definitions

Follow these steps:

1. Click **Navigator > Setup and Maintenance**.
2. In the Setup and Maintenance work area, go to the following:
   - Functional Area: Application Extensions or a product-specific functional area
   - Task: Manage Enterprise Scheduler Job Definitions and Job Sets
3. On the Manage Enterprise Scheduler Job Definitions and Job Sets page, open the Manage Job Definitions tab.
4. Select the job definition you want to edit.
5. Click **Edit**.
6. Make the changes that you want, for example:
   - You can edit the display name of the job definition to use terms that are more familiar to your users.
   - You can use the **Prompt** field to edit parameter display names.
7. Click **Save and Close**.

Predefined Job Definitions

You cannot update parameters in predefined job definitions, but this table lists some of the other fields that you can edit.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retries</td>
<td>The number of times to automatically run this job again if the scheduled process fails.</td>
</tr>
<tr>
<td>Job Category</td>
<td>Specific to the application of the job definition, it's used to group definitions according to your requirements.</td>
</tr>
<tr>
<td>Timeout Period</td>
<td>The amount of time before stopping a scheduled process that couldn't complete.</td>
</tr>
<tr>
<td>Priority</td>
<td>Priority of scheduled processes submitted, with 0 as lowest. If other processes, based on the same or another job, are scheduled to run at the same time, then priority determines the run order.</td>
</tr>
</tbody>
</table>

Define Parameters for Job Definitions

A parameter controls which records are included or how they are affected when a job runs. **Job definitions** can have one or more parameters or none at all. You define parameters while creating or editing job definitions using the Manage Enterprise Scheduler Job Definitions and Job Sets page. In the Setup and Maintenance work area, use the following:

- Functional Area: Application Extensions or a product-specific functional area
- Task: Manage Enterprise Scheduler Job Definitions and Job Sets
When users run the scheduled process, the values they enter for the parameters determine the data to be included in the report. Also, the values are passed to the data model that the report is using.

The parameters that you define must be in the same order as parameters in the data model. For example, the data model has parameters in this order:

- P_START_DATE
- P_END_DATE
- P_CURRENCY

You create parameters as follows:

- Start Date
- End Date
- Currency

### Defining Parameters: Job Definitions

To define parameters while creating or editing job definitions:

1. On the Manage Job Definitions page, open the Parameters sub tab.
2. Click **Create**.
3. Enter the parameter prompt that users see when they submit the scheduled process.
4. Select a data type and configure how the parameter and the data entered are displayed, as described in this table.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>Select this if you want the parameter to be a check box.</td>
</tr>
<tr>
<td></td>
<td>Select <strong>True</strong> or <strong>False</strong> to determine if the check box is selected</td>
</tr>
<tr>
<td></td>
<td>or not.</td>
</tr>
<tr>
<td>Date or time</td>
<td>Select <strong>Date and time</strong> or <strong>Date only</strong> option.</td>
</tr>
<tr>
<td></td>
<td>Select a value from the <strong>Default Date Format</strong>.</td>
</tr>
<tr>
<td>Number</td>
<td>Select a <strong>Number Format</strong>.</td>
</tr>
<tr>
<td></td>
<td>Select <strong>Left</strong> or <strong>Right</strong> for data alignment.</td>
</tr>
<tr>
<td>String</td>
<td>Select a <strong>Page Element</strong>.</td>
</tr>
<tr>
<td></td>
<td>Select <strong>Text box</strong> if you want the user to provide a text.</td>
</tr>
<tr>
<td></td>
<td>Select <strong>Choice list</strong> if you want a list with limited options (maximum 10).</td>
</tr>
<tr>
<td></td>
<td>Select <strong>List of values</strong> if you want a list with unlimited options with a search facility.</td>
</tr>
</tbody>
</table>

5. Select the **Read Only** check box if you don't want to enable users to set this parameter. When a parameter is set as read only, the user is required to provide a default value to be passed to the job definition.
If you select list of values or choice list page element, select a **List of Values Source** and an **Attribute**.

From the list of available attributes, select the attributes you want to appear in the list and move them to the selected attributes section. These attributes determine the values that the user can see.

Define a **Default Value** for the parameter.

In the **Tooltip Text** field, provide additional information for the user to follow.

Select the **Required** check box if users must set this parameter to submit the scheduled process.

Select the **Do not Display** check box if users should not see this parameter while submitting the process.

Click **Save and Create Another** or **Save and Close**.

**Dependent Parameters**

The attributes of some parameters depend on the attributes or values of certain other parameters. The attributes of a parameter would change if the value of its dependent parameter changes.

For example, you have three parameters, namely Country, State and, City. In this case, the value of the Country parameter would determine the values available in the State parameter. The values in the State parameter would determine the values available in the City parameter.

**Define Dependent Parameters in Job Definitions**

This example demonstrates how to define dependent parameters.

This table summarizes key decisions for this scenario.

<table>
<thead>
<tr>
<th>Decisions to Consider</th>
<th>In this Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Which parameters do I want to make dependent?</td>
<td><strong>Region, Country</strong> and <strong>City</strong>.</td>
</tr>
<tr>
<td><strong>Region</strong> list of values includes the names of the regions like North America, EMEA, JAPAC and so on. The <strong>Country</strong> list of values includes the names of countries like USA, Canada, France, England, India, Japan and so on. Similarly <strong>City</strong> list of values includes the names of different cities like New York, Washington, London, Paris, Mumbai, Tokyo and so on. The <strong>Country</strong> parameter list of values includes only the countries like USA, Canada and so on, if user selects North America. The <strong>City</strong> parameter list of values includes the names of the cities in the country that the user has selected.</td>
<td></td>
</tr>
<tr>
<td>What are view criteria?</td>
<td>The view criteria determine the values that appear in the dependent parameter list of values for the users to see. The view criteria are used to filter the list and pass the required bind variables based on the user's selection. To filter countries based on the selected region, you must select <strong>getcountriesbyRegion</strong> and pass <strong>Region</strong> as a bind variable.</td>
</tr>
</tbody>
</table>
Prerequisites

Create the parameters Region, Country and City. The values available to the users in the Country parameter list of values depends on the value selected for the Region parameter. City parameter list of values depends on the value that the user selects for the Country parameter.

Defining Dependent Parameters

To define parameters with dependent lists of values:

1. On the Manage Job Definitions tab, open the Parameters sub tab.
2. Select the Region parameter.
3. Click the Manage Dependencies button located next to the Delete button.
4. From Available View Criteria, select getCountriesByRegion and move it to Selected View Criteria using the move icons. The selected view criteria appears in the Bind Variables section.
5. In the Bind Variables section, for the getCountriesByRegion view criteria, select Country from the mapped parameters list of values. The Country parameter list of values is now dependent on the value selected for the Region parameter.
6. Click OK.
7. Repeat the steps with Country parameter. Select getCitiesByCountries from the available view criteria and pass City as a bind variable.

Job Sets

Job Sets

A job set is a collection of several jobs in a single process set that the users can submit instead of running individual jobs separately. The job set definition also determines if the jobs run in serial or parallel, or based on some other predetermined logic. In the Setup and Maintenance work area, go to the following:

- Functional Area: Application Extensions or a product-specific functional area
- Task: Manage Enterprise Scheduler Job Definitions and Job Sets

Use the Manage Enterprise Scheduler Job Definitions and Job Sets page to open the Manage Job Sets tab.

- On this tab, you can view and define job sets, and use Query By Example to find a specific job set.

You can't edit or delete the predefined job sets which are indicated by an asterisk. You can create job sets, and also edit and delete job sets that are not predefined.

Job Set Steps

A job set can contain any number of individual jobs as well as other job sets. There can also be multiple levels of nested job sets within a single job set. For example, a job set can include three jobs and two job sets, one of which contains another job set. Each individual job or job set that's included within a job set is called a job set step. A job set and each of its job set steps can have additional parameters. Users provide the values for these parameters when they submit the process set.
Application Defined Properties

Some Job Sets require the user to enter certain additional properties before submitting a job request. Application Defined Properties are the additional properties that the user has to enter during runtime. For example, when `enableDuplicateJobWithParamTaskflow` property is set to **True**, it can be used to run a single job multiple times with different parameter values, within a job set.

System Properties

System Properties are the additional settings that determine how a job set runs. For example, you can use a system property to determine the number of retries for a job set that fails to execute. On this tab, you can view and define job sets, and use Query By Example to find a specific job set.

Create Job Sets

Create a job set so that users can run multiple jobs with a single submission. Before you create the job set, plan the sequence and hierarchy of the job steps within the job set. You can create job sets using the Manage Job sets tab. You can also edit and delete job sets that are not predefined.

Steps to create Job Sets

Follow these steps:

1. Click **Navigator > Setup and Maintenance**.
2. In the Setup and Maintenance work area, go to the following:
   - Functional Area: Application Extensions a product-specific functional area
   - Task: Manage Enterprise Scheduler Job Definitions and Job Sets
3. On the Manage Enterprise Scheduler Job Definitions and Job Sets page, open the Manage Job Sets tab.
4. Click **Create**.
5. Complete the fields as shown in this table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Provide a name with only alphanumeric characters, for example, ExportMetadataTables1.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> A job set name can't have space or any special character.</td>
</tr>
<tr>
<td>Display Name</td>
<td>Provide a name that the users see while submitting the scheduled process.</td>
</tr>
<tr>
<td>Description</td>
<td>Provide more information about what the job set does.</td>
</tr>
<tr>
<td>Package</td>
<td>Specify the path where you want to save the job set.</td>
</tr>
</tbody>
</table>

6. In the Job Set Steps section, select **Serial** or **Parallel** to define the sequence of the job set.
7. Click **Add Job Set Step** to open the Add Step dialog box.
8. In the Step tab:
   a. Enter a **Step ID** to easily identify the steps.
   b. Search for and select a job or job set to include in your job set.
   c. Select **Job definition** or **Job set**.
   d. Type a valid name or package, or both of the Job or Job Set you are looking for and click search.
   e. Select the required job definition or job set and click **OK**.
   f. If you selected Parallel in step 4:
      • Select **Insert into main diagram** if you want the step to be executed independently.
      • Select **Add to list of available steps** if you want the step to be one of the outcomes in the available steps.
      • If you choose to add the step to the list of available steps, select an option for the possible job outcomes. For example, you can determine whether the process must stop or another step must be executed if the step fails to run successfully.

9. In the **Application Defined Properties** tab:
   a. Click **Add Application Defined Property** icon and select a data type.
   b. Enter a name and an initial value.
   c. Select the **Read Only** check box if you don’t want users to update this property when they submit the process set.
   d. Click **OK**.

10. In the **System Properties** tab:
    a. Click **Add System Property**.
    b. From the name list, select a system property.
    c. Enter a value in the **Initial Value** field.
    d. Select **Read Only** check box if you don’t want users to update this property when they submit the process set.
    e. Click **OK**.

    **Note:** You can also add and edit the Application Defined Properties and System Properties in the respective sections on the Create Job Set page.

11. Click **OK** to add the Job Set Step.
12. Add jobs and job sets as required. You can select job steps in the Job Set Steps section and edit, remove, or reorder (for Serial only). You can also switch between **Serial** and **Parallel**.
13. Click **Save and Close**.

    **Note:** When you create a Job Set, the **privilege** of the same name as the Job Set is automatically created. For example, for a Job Set named ExportAppsData, the privilege is named **RUN_EXPORTAPPSDATA**.

### System Properties for Job Sets

System Properties are the additional settings that determine how a job set runs. For example, you can use a system property to determine the number of retries for a job set that fails to execute. You can open the Manage Job Sets tab using the Manage Enterprise Scheduler Job Definitions and Job Sets page. In the Setup and Maintenance work area, use the following:

- **Functional Area:** Application Extensions or a product-specific functional area
- **Task:** Manage Enterprise Scheduler Job Definitions and Job Sets
System Properties

This table lists some system properties with description.

<table>
<thead>
<tr>
<th>System Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYS_* allowMultPending</td>
<td>Specifies if the same job definition can have multiple pending requests.</td>
</tr>
<tr>
<td>SYS_* application</td>
<td>Specifies the logical name of the Scheduling Services folder application used for request processing. Oracle Enterprise Scheduler automatically sets this property during request submission.</td>
</tr>
<tr>
<td>SYS_* effectiveApplication</td>
<td>Specifies the logical name of the Scheduling Services folder application that is the effective application used to process the request. You can associate a job definition, job type, or a job set step with a different application by defining the EFFECTIVE APPLICATION system property. This property can only be specified through metadata and cannot be specified as a submission parameter.</td>
</tr>
<tr>
<td>SYS_* priority</td>
<td>Specifies the request processing priority. The priority interval is 0 to 9, where 0 is the lowest priority and 9 is the highest. If this property is not specified, the default value used is 4.</td>
</tr>
<tr>
<td>SYS_* product</td>
<td>Specifies the product used for submitting the request.</td>
</tr>
<tr>
<td>SYS_* request_timeout</td>
<td>Enables the job request to time out.</td>
</tr>
<tr>
<td>SYS_* requestExpiration</td>
<td>Specifies the expiration time for a request. This represents the time (in minutes) that a request will expire after its scheduled execution time. An expiration value of zero (0) means that the request never expires. If this property is not specified, the default value used is 0. Request expiration only applies to requests that are waiting to run. If a request waits longer than the specified expiration period, it does not run. After a request starts running, the request expiration no longer applies.</td>
</tr>
<tr>
<td>SYS_* retries</td>
<td>Specifies the retry limit for a failed request. If request execution fails, the request is retried up to the number of times specified by this property until the request succeeds. If the retry limit is zero (0), a failed request is not retried. If this property is not specified, the default value used is 0.</td>
</tr>
</tbody>
</table>
10 Define Applications Core Configuration

Overview of Applications Core Configuration

The Define Applications Core Configurations task list contains the Oracle Middleware Extensions for Oracle Applications (Applications Core) tasks that support implementation of common functionality such as lookups, profile options, document sequences, and so on. Some of the tasks are also available in the Application Extensions functional area. You may also find specific versions of this task list depending upon the product family or the offering that uptakes those tasks.

Use this task list to manage configuration objects that are defined centrally and shared across applications, in addition to tasks classified in the Maintain Common Reference Objects task list. You can search for this task list in the Setup and Maintenance work area.

Related Topics
- Overview of Common Reference Objects

Import Flexfields, Lookups, or Profile Values Using Web Services

You can import lookups, profile option values, and extensible flexfields into an application using the import function on the application page. Alternatively, you can use the web service FndManageImportExportFilesService for the import task. The web service uses the following methods:

<table>
<thead>
<tr>
<th>Loader Name</th>
<th>Asynchronous Method</th>
<th>Synchronous Methods</th>
<th>Supported methods for backward compatibility (without UUID)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flex data uploader</td>
<td>processFlexDataAsync(List&lt;String&gt; inputFileTypeList, String UUID)</td>
<td>processFlexData(List&lt;String&gt; inputFileTypeList, String UUID)</td>
<td>processFlexDataFiles(List&lt;String&gt; inputFileTypeList)</td>
</tr>
<tr>
<td>Lookups data uploader</td>
<td>processLookupsDataAsync(long lookupTypesFileID, long lookupCodesFileID, long viewApplicationID, String UUID)</td>
<td>processLookupsData (long lookupTypesFileID, long lookupCodesFileID, long viewApplicationID, String UUID)</td>
<td>processLookupsUploadFiles (long lookupTypesFileID, long lookupCodesFileID, long viewApplicationID)</td>
</tr>
<tr>
<td>User profile values uploader</td>
<td>processUserProfileValuesAsync(long userProfileFileID, String UUID)</td>
<td>processUserProfileValues(long userProfileFileID, String UUID)</td>
<td>processUserProfileValuesFile (long userProfileFileID)</td>
</tr>
</tbody>
</table>
Example

The following example is about importing the profile values using the web service. To import lookups or extensible flexfields, initiate the web service in the same way as shown here, after replacing the methods and other relevant values in the code.

To import profile values:

1. Perform the following substeps to generate keystore for the security policy "oracle/wss11_saml_or_username_token_with_message_protection_service_policy":
   a. Generate keystore using the command `keytool -genkeypair -keyalg RSA -alias mycompkey -keypass password -keystore mycompclient-keystore.jks -storepass password -validity 3600`
   b. Open the WSDL using the URL `http://<host>:<port>/fndAppCoreServices/FndManageImportExportFilesService?wsdl` through a web browser, and get the public key `<wsdl:service>/<wsdl:port>/<wsa:EndpointReference>/<wsid:Identity>/<dsig:keyInfo>/<dsig:X509Data>/<dsig:X509Certificate>`. Then, enclose it with `---- BEGIN CERTIFICATE -----` and `---- END CERTIFICATE ----` and save it to a file by name cdrmpk.cer.
   c. Store the key information in the truststore using the command `keytool -importcert -alias cdrmkey -file cdrmpk.cer -keystore mycompclient-keystore.jks -storepass password`.
2. Open the WSDL page using the browser of your choice.
3. Export and save the associated security certificates Verisign Class 3 Public Primary Certification Authority - G5 and Verisign Secure Server CA - G3.
4. Use the following command to import the saved certificates into the trust store of the client computer.
   `keytool -importcert -keystore <truststore> -storepass <truststorepassword> -file <file location where the mentioned certificate is stored> -alias <alias for certificate>`
5. Run the following command to generate the JAX-WS proxy for the `FndManageImportExportFilesService` web service.
   `C:\Program Files\Java\jdk1.7.0_04\bin\wsimport -s "d:\wsimport\FndManageImportExport" -d "d:\wsimport\FndManageImportExport" http://<host>:<port>/fndAppCoreServices/FndManageImportExportFilesService?wsdl`
6. Save the generated code as a JAR file and name it `FndManageImportExportProxy.jar`.
7. Use the following code to create another JAR file to initiate the web service:
   ```java
   package com.oracle.xmlns.oracle.apps.fnd.applcore.webservices;
   import com.sun.xml.ws.developer.WSBindingProvider;
   ```

<table>
<thead>
<tr>
<th>Loader Name</th>
<th>Asynchronous Method</th>
<th>Synchronous Methods</th>
<th>Supported methods for backward compatibility (without UUID)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valueset Values data uploader</td>
<td><code>processValueSetValuesAsync(Long fileNameAtRepository, String UUID)</code></td>
<td><code>processValueSetValues(Long fileNameAtRepository, String UUID)</code></td>
<td><code>valueSetValuesDataLoader(Long fileNameAtRepository)</code></td>
</tr>
<tr>
<td>Generic method</td>
<td>NA</td>
<td><code>downloadProcessLogFile(String UUID)</code></td>
<td>NA</td>
</tr>
</tbody>
</table>
import java.io.File;
import java.io.IOException;
import java.util.List;
import java.util.Map;
import javax.xml.ws.BindingProvider;
import javax.xml.ws.WebServiceRef;
import javax.xml.ws.handler.Handler;
import oracle.webservices.ClientConstants;
import weblogic.wsee.jws.jaxws.owsm.SecurityPoliciesFeature;
// !THE CHANGES MADE TO THIS FILE WILL BE DESTROYED IF REGENERATED!
// This source file is generated by Oracle tools
// Contents may be subject to change
// For reporting problems, use the following
// Version = Oracle WebServices (11.1.1.0.0, build 130224.1947.04102)

public class FndManageImportExportFilesServiceSoapHttpPortClient {
    @WebServiceRef
    private static FndManageImportExportFilesService_Service fndManageImportExportFilesService_Service;

    public static void main(String[] args) {
        System.setProperty("javax.net.ssl.trustStore","<location of truststore which is used in II to import the certificates>");
        System.setProperty("javax.net.ssl.trustStorePassword", "<truststore password>");

        fndManageImportExportFilesService_Service = new FndManageImportExportFilesService_Service();
        FndManageImportExportFilesService fndManageImportExportFilesService = fndManageImportExportFilesService_Service.getFndManageImportExportFilesServiceSoapHttpPort(securityFeatures);
        // Add your code to call the required methods.

        WSBindingProvider wsbp = (WSBindingProvider) fndManageImportExportFilesService;
        Map<String, Object> requestContext = wsbp.getRequestContext();
        requestContext.put(ClientConstants.WSSEC_KEYSTORE_TYPE, "jks");
        // Provide location of 'mycompclient-keystore.jks' which was created during Step I)
        requestContext.put(ClientConstants.WSSEC_KEYSTORE_LOCATION, "<home/user1/mycompclient-keystore.jks>");
        requestContext.put(ClientConstants.WSSEC_KEYSTORE_PASSWORD, "password");
        requestContext.put(ClientConstants.WSSEC_RECIPIENT_KEY_ALIAS, "cdrmkey");
        //Provide user who is having permission to initiate the service
        requestContext.put(WSBindingProvider.USERNAME_PROPERTY, "<username>");
        requestContext.put(WSBindingProvider.PASSWORD_PROPERTY, "<password>");

        String id = invokeUploadFiletoUCMMethod(fndManageImportExportFilesService);
        if (id != null) {
            invokeUserProfileValuesDataLoader(fndManageImportExportFilesService, new Long(id));
        }
    }
}
static String invokeUploadFiletoUCMMethod(FndManageImportExportFilesService fndManageImportExportFilesService) {

    String response = null;
    DocumentDetails document = new DocumentDetails();
    ObjectFactory objfactory = new ObjectFactory();
    document.setFileName(objfactory.createDocumentDetailsFileName("import_data.txt"));
    // Provide UCM repository - if repository is fin/tax/import then suffix each value with $ as mentioned here
    document.setDocumentTitle(objfactory.createDocumentDetailsDocumentTitle("VS"));
    document.setContentType(objfactory.createDocumentDetailsContentType("plain/text"));

    try {
        // Provide location of bulk import data file in prescribed format
        byte[] content = org.apache.commons.io.FileUtils.readFileToByteArray(new File("/home/user1/import_data.txt"));
        // System.out.println("File content:" + new String(content, "UTF-8"));
        document.setContent(objfactory.createDocumentDetailsContent(content));

        // Provide UCM repository - if repository is fin/tax/import then suffix each value with $ as mentioned here
        document.setDocumentTitle(objfactory.createDocumentDetailsDocumentTitle("VS"));
        document.setContentType(objfactory.createDocumentDetailsContentType("plain/text"));

        try {
            response = fndManageImportExportFilesService.uploadFiletoUCM(document);
            System.out.println("Response: " + response);
        } catch (ServiceException e) {
            System.out.println(e.getMessage());
        }
    }
    catch (Exception e) {
        System.out.println("Exception: "+e.getMessage());
    }

    return response;
}

8. Save the generated output as a JAVA file and name it FndManageImportExportFilesServiceSoapHttpPortClient.java.

9. Use the JAVA file to build a JAR file, and name it FndManageImportExportClient.jar.

10. Use the following command to run the web service:

    Note: Wherever applicable, replace the values referring to the profile values with those of the lookups or flexfields value sets.
Here's a sample program that's a web service client for the asynchronous method `deployFlexAsync()`. This client program starts callback client and does the asynchronous call. The required callback web service starts at the line no 58.

```java
java -cp ./FndManageImportExportProxy.jar:./ws.api_1.1.0.0.jar:/FndManageImportExportClient.jar
FndManageImportExportFilesServiceSoapHttpPortClient
```

The callback web service will still be in Running state after the client program execution completes and you need to close it manually. Alternatively, you can also comment the previous line here and run it in a separate client program.

```java
package com.ws.client;
import com.sun.xml.ws.api.addressing.AddressingVersion;
import com.sun.xml.ws.api.addressing.WSEndpointReference;
import com.sun.xml.ws.developer.WSBindingProvider;
import com.sun.xml.ws.message.StringHeader;
import com.ws.client.callback.ApplicationsCoreSetupServiceResponseImpl;
import com.ws.client.types.DeployFlexAsyncResponse;
import java.util.Map;
import java.util.UUID;
import javax.xml.ws.Endpoint;
import javax.xml.ws.Response;
import javax.xml.ws.WebServiceRef;
import oracle.webservices.ClientConstants;
import weblogic.wsee.jws.jaxws.owsm.SecurityPolicyFeature;

public class ApplicationsCoreSetupServiceSoapHttpPortClient
{

private static final AddressingVersion WS_ADDR_VER = AddressingVersion.W3C;

public static void main(String [] args) throws InterruptedException {
    applicationsCoreSetupService_Service = new ApplicationsCoreSetupService_Service();
    SecurityPolicyFeature[] features = new SecurityPolicyFeature[]
    { new SecurityPolicyFeature("policy:oracle/wss_username_token_client_policy")
    };

    ApplicationsCoreSetupService applicationsCoreSetupService =
    applicationsCoreSetupService_Service.getApplicationsCoreSetupServiceSoapHttpPort(features);

    // Get the request context to set the outgoing addressing properties
    WSBindingProvider wsbp = (WSBindingProvider)applicationsCoreSetupService;
    Map<String, Object> requestContext = wsbp.getRequestContext();

    requestContext.put(ClientConstants.WSSEC_KEYSTORE_TYPE, "jks");
    requestContext.put(ClientConstants.WSSEC_KEYSTORE_LOCATION,
        "/scratch/vgarikip/view_storage/work/keys/mycompclient-keystore.jks");
    requestContext.put(ClientConstants.WSSEC_KEYSTORE_PASSWORD,
        "mypassword1");
```
requestContext.put(ClientConstants.WSSEC_RECIPIENT_KEY_ALIAS, "cdrmkey");
requestContext.put(WSBindingProvider.USERNAME_PROPERTY, "app_impl_consultant");
requestContext.put(WSBindingProvider.PASSWORD_PROPERTY, "mypassword1");

String callbackServiceAddress = "http://<server>:<port>/TestAsyncMani-Project1-context-root/
ApplicationsCoreSetupServiceResponseImplPort";
Endpoint e = Endpoint.publish(callbackServiceAddress, new ApplicationsCoreSetupServiceResponseImpl());

WSEndpointReference replyTo = new WSEndpointReference(callbackServiceAddress, WS_ADDR_VER);
String uuid = "uuid:" + UUID.randomUUID();
wbp.setOutboundHeaders( new StringHeader(WS_ADDR_VER.messageIDTag, uuid),
replyTo.createHeader(WS_ADDR_VER.replyToTag));

applicationsCoreSetupService.deployFlexAsync("PER_CITIZENSHIPS_DFF", "DFF", false); /*.deployPatchedFlexAsync(mode, pCustomizationId, pCustomizationSetLoc);

System.out.println("####END OF WS CALL");
Thread.sleep(10000);

// Add your code to call the desired methods.

Example callback web service start utility if you comment line no 58 in the previous program:

package com.ws.client;

import com.ws.client.callback.ApplicationsCoreSetupServiceResponseImpl;
import javax.xml.ws.Endpoint;

public class RunCallbackServtest {
    public RunCallbackServtest() {
        super();
    }

    public static void main(String[] args) throws InterruptedException {
        RunCallbackServtest runCallbackServtest = new RunCallbackServtest();
        runCallbackServtest.runServ();
    }

    private void runServ() throws InterruptedException {
        System.out.println("$$$$$ BEFORE Server Start
ApplicationsCoreSetupServiceResponseImplPort":
        System.out.println("$$$$$ AFTER Server Start");
    }
}
Define Lookups

Overview of Lookups

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.

The following table contains an example of a lookup type for marital status (MAR_STATUS) that has lookup codes for users to specify married, single, or available legal partnerships.

<table>
<thead>
<tr>
<th>Lookup Code</th>
<th>Meaning</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Married</td>
<td>Not applicable</td>
</tr>
<tr>
<td>S</td>
<td>Single</td>
<td>Not applicable</td>
</tr>
<tr>
<td>R</td>
<td>Registered Partner</td>
<td>+NL</td>
</tr>
<tr>
<td>DP</td>
<td>Domestic Partner</td>
<td>-FR, AU</td>
</tr>
</tbody>
</table>

In this case, tags are used for localizing the codes. All legislations list Married and Single. Only the Dutch legislation lists Registered Partner. And all legislations except France and Australia also list Domestic Partner.

When managing lookups, you need to understand the following.

- Using lookups in applications
- Configuration 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.

**Configuration Level**

The configuration 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 configuration 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 predefined lookup codes can't be modified. Some predefined lookup codes can be changed during implementation or later, as needed.

The configuration levels are user, extensible, and system. The following table shows the lookup management tasks permitted at each configuration level.

<table>
<thead>
<tr>
<th>Permitted Task</th>
<th>User</th>
<th>Extensible</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deleting a lookup type</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Inserting new codes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Updating start date, end date, and enabling the lookup code</td>
<td>Yes</td>
<td>Yes, only if the code isn't predefined data</td>
<td>No</td>
</tr>
<tr>
<td>Deleting codes</td>
<td>Yes</td>
<td>Yes, only if the code isn't predefined data</td>
<td>No</td>
</tr>
<tr>
<td>Updating tags</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Updating module</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Predefined data means LAST_UPDATED_BY = SEED_DATA_FROM_APPLICATION.

If a product depends on a lookup, the configuration level must be system or extensible to prevent deletion.

Once the configuration level is set for a lookup type, it can't be modified. The configuration level for newly created lookup types is by default set at the User level.

**Standard, Common, and Set-Enabled Lookups**

The following table shows the available types of lookups.

<table>
<thead>
<tr>
<th>Lookup Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>Lists the available codes and translated meanings.</td>
</tr>
</tbody>
</table>
Standard lookups are the simplest form of lookup types consisting only of codes and their translated meaning. They differ from common lookups only in being defined in the standard lookup view. Common lookups exist for reasons of backward compatibility and differ from standard lookups only in being defined in the common lookup view. These can also be lookups having attribute columns. 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 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.

To view the predefined lookups and their lookup codes, use the following tasks in the Setup and Maintenance work area:

- Manage Standard Lookups
- Manage Common Lookups
- Manage Set-Enabled Lookups

### Translating Lookups

You can translate the lookups that you defined to the preferred language(s) without changing the language session of the application. Use the translation option available on the lookup code table. By default, for each lookup, all the permitted language rows in the translator dialog box appear in the source language (the current session language).

When you edit a particular language entry, you can modify the translated meaning and description to the language in which you want the lookup to appear. Once the updates are made, the end-users can view the lookup in the translated text.
Note: You can add the translation for only as many languages as are permitted by the administrator. The functionality to limit the number of languages displayed on the dialog box is controlled through the Translation Editor Languages profile option. It can be set at the SITE or USER level. If nothing is specified, all active languages are displayed.

Related Topics
- Enter or Edit Translated Text

Example of a Standard Lookup

Creating a new standard lookup involves creating or selecting a lookup type containing the lookup code. The task also involves determining appropriate values for the lookup codes and their meanings. You can only create or edit lookup codes for a particular lookup type if its configuration level supports it.

Creating a Lookup Type Called COLORS

Your enterprise needs a list of values to be used as different statuses on a process. Each status is indicated using a color. Therefore, you create a lookup type called COLORS. The following table lists a mapping between the lookup type parameters and the actual values assigned to those parameters to create the required list of values.

<table>
<thead>
<tr>
<th>Lookup type parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup type name</td>
<td>COLORS</td>
</tr>
<tr>
<td>Meaning</td>
<td>Status</td>
</tr>
<tr>
<td>Description</td>
<td>Status by color</td>
</tr>
<tr>
<td>Module</td>
<td>Oracle Fusion Middleware Extensions for Oracle Application</td>
</tr>
</tbody>
</table>

After you define the lookup type, you need to define the lookup codes and their related details. The following table lists the lookup codes you define for the COLORS lookup type.

<table>
<thead>
<tr>
<th>Lookup Code</th>
<th>Meaning</th>
<th>Enabled</th>
<th>Display Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLUE</td>
<td>Urgent</td>
<td>No</td>
<td>4</td>
</tr>
<tr>
<td>RED</td>
<td>Stop</td>
<td>Yes</td>
<td>1</td>
</tr>
<tr>
<td>GREEN</td>
<td>Proceed</td>
<td>Yes</td>
<td>3</td>
</tr>
<tr>
<td>YELLOW</td>
<td>Check</td>
<td>Yes</td>
<td>2</td>
</tr>
</tbody>
</table>
The Resulting Data Entry List of Values

Only the enabled lookup codes appear in the list of values for the COLORS lookup type. You must select one of them to complete the activity.

The following table lists the meanings and the codes that were enabled. They appear in the order of the defined display sequence.

<table>
<thead>
<tr>
<th>Meaning</th>
<th>Lookup Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop</td>
<td>RED</td>
</tr>
<tr>
<td>Check</td>
<td>YELLOW</td>
</tr>
<tr>
<td>Proceed</td>
<td>GREEN</td>
</tr>
</tbody>
</table>

Analysis

The BLUE lookup code wasn't enabled and doesn't appear in the list of values. The display sequence of values in the list of values is alphabetic, unless you enter a number manually to determine the order of appearance. Number 1 indicates the first value that appears in the list. Only lookups that are enabled and active between start and end dates are visible.

The Transaction Table

When users enter one of the values from the list of values for the lookup type COLORS, the transaction table records the lookup code. The following table contains an example, where the lookup code is stored in the Status column of the transaction table.

<table>
<thead>
<tr>
<th>Transaction number</th>
<th>User name</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jane</td>
<td>RED</td>
</tr>
<tr>
<td>2</td>
<td>Bob</td>
<td>YELLOW</td>
</tr>
<tr>
<td>3</td>
<td>Alice</td>
<td>BLUE</td>
</tr>
</tbody>
</table>

The status for one user is BLUE because at the time they entered a value, BLUE was enabled. Disabling a lookup code doesn't affect transaction records in which that code is stored. Data querying and reporting have access to disabled lookup codes in transaction tables.
Example of a Set-Enabled Lookup

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. You can only create or edit lookup codes for a particular lookup type if its configuration 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. Use the Manage Set Assignments task to define and manage reference data set assignments.

Selecting a Reference Group for a Set-Enabled Lookup Type

Specify a reference group for a set-enabled lookup type to 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. For example, there are two references data sets - one for the US and the other for EU. If a COLORS lookup type contains RED, YELLOW, ORANGE, and GREEN lookup codes, you can enable one RED lookup code from the US reference data set and another RED lookup code from the EU reference data, each lookup code having different meanings.

The following table elaborates the example, how these two reference data sets (US and EU) contain one lookup code that's common, but each differing in its lookup meaning.

<table>
<thead>
<tr>
<th>Reference Data Set</th>
<th>Lookup Code</th>
<th>Lookup Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>RED</td>
<td>Red</td>
</tr>
<tr>
<td>US</td>
<td>YELLOW</td>
<td>Yellow</td>
</tr>
<tr>
<td>US</td>
<td>GREEN</td>
<td>Green</td>
</tr>
<tr>
<td>EU</td>
<td>RED</td>
<td>Rouge</td>
</tr>
<tr>
<td>EU</td>
<td>ORANGE</td>
<td>Orange</td>
</tr>
</tbody>
</table>

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 the example.

In another example in the following table, a lookup type called HOLD_REASON provides a list of reasons for putting a contract renewal on hold. Reference data sets determine which codes are included in the Hold Reason list of values.

<table>
<thead>
<tr>
<th>Reference Data Set</th>
<th>Lookup Code</th>
<th>Lookup Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>SEC</td>
<td>SEC Compliance Review</td>
</tr>
<tr>
<td>Reference Data Set</td>
<td>Lookup Code</td>
<td>Lookup Meaning</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
<td>----------------------------------------------------</td>
</tr>
<tr>
<td>US</td>
<td>DIR</td>
<td>Needs Director’s Approval</td>
</tr>
<tr>
<td>US</td>
<td>VP</td>
<td>Needs Vice President’s Approval</td>
</tr>
<tr>
<td>CHINA</td>
<td>CSRC</td>
<td>Pending China Securities Regulatory Commission Review</td>
</tr>
<tr>
<td>CHINA</td>
<td>PR</td>
<td>Needs President’s Approval</td>
</tr>
<tr>
<td>COMMON</td>
<td>REQUESTED</td>
<td>Customer Request</td>
</tr>
</tbody>
</table>

Referring to the example in the table, when end-users place a contract on hold in the US business unit, the three reason codes in the US set are available. When placing a contract on hold in the China business unit, the two codes in the China set are available.

**Import Lookups**

On each page pertaining to the tasks of managing the Standard, Common, and Set Enabled lookups, use the Import option to import the lookup type and lookup code information.

**Prerequisite**

The separate files containing the lookup types and lookup codes are already available in the document repository of Oracle WebCenter Content.

**Importing Lookups**

To import lookups:

1. In the Setup and Maintenance work area, go to the Manage Standard Lookups task. Depending on the lookup you want to import, you may select the other lookup tasks.
2. In Search Results, from the Actions menu, select Import.

   The Import Lookups dialog box appears.
3. Select the WebCenter Content account to which the files were uploaded.
4. Enter the names of the separate files containing the lookup type and lookup code information. The names here must match with the names of the files uploaded to the selected account.
5. Click Upload. The lookup details are imported.

   **Note:** If the import fails, click the link to the log file on the confirmation dialog box and examine the cause of failure.
File Format for Importing Lookups

To import lookups into an application, you create separate text files containing the lookup types and lookup codes and upload them to the Oracle WebCenter Content document repository. The files must follow a specific format, as described here. After the files are in the document repository, you can then import the lookup types and lookup codes into the application following the instructions in the Importing Lookups: Procedure topic.

While creating the file, adhere to the following guidelines:

- Use a vertical bar or pipe ( | ) as a delimiter between fields for both header and value rows.
- Set the file encoding to UTF-8 without the Byte Order Mark (BOM), as per the Oracle WebCenter Content specification.

The following sections contain details about the specific lookup types and codes.

Prerequisite

You must have worked with lookups in Oracle Cloud applications.

Standard and Common Lookups

The lookup types and codes are similar for standard and common lookups. To create a file containing the lookup types, include the following headers:

- **LookupType**: The lookup type.
- **Meaning**: The display name of the lookup type.
- **Description**: The description of the lookup type. This header is optional.
- **ModuleType**: The module with which the lookup type is associated.
- **ModuleKey**: The module code.

Here's a sample file that contains the header values at the beginning of the file, followed by line entries of the two lookup types that are to be imported. For importing several lookup types, add more line entries in a similar format.

```
LookupType|Meaning|Description|ModuleType|ModuleKey
AFLOG_22APR_1|Log1|AFLOG_desc_1|APPLICATION|FND
PROD_22APR_2|Product1|PROD_desc_2|APPLICATION|FND
```

To create a file containing the lookup codes, include the following headers.

- **Required headers**:
  - **LookupType**: The lookup type.
  - **LookupCode**: The lookup code associated with the lookup type.
  - **DisplaySequence**: The sequence position at which the lookup code appears in the list of values.
  - **EnabledFlag**: Indicates the status of the lookup code, whether it's enabled for display or not.
  - **Meaning**: The display name of the lookup code.

- **Optional headers**:
  - **StartDateActive**: Beginning of the date range during which the lookup code is active and visible on the page. The format is dd/M/yyyy.
End Date Active: End of the date range during which the lookup code is active and visible on the page. The format is dd/M/yyyy.

Description: Description of the lookup code.

Tag: Any tag associated with the lookup code that may be used for a quick reference or retrieval of information.

Seg: The name of the API used for a global segment defined for the descriptive flexfield associated with the lookup.

CONTEXT ATTRIBUTE: The context value specified in the CSV file for a particular descriptive flexfield.

ctxSeg: The name of the API for a context sensitive segment defined for a context attribute.

Here's a sample file that contains two lookup codes:

```csv
LookupType|LookupCode|DisplaySequence|EnabledFlag|StartDateActive|EndDateActive|Meaning|Description|Tag|Seg|
CONTEXT_ATTRIBUTE|ctxSeg
TASK_22APR_1|Code1_1|1|Y|25/12/2014|25/5/2015|TASK_22apr_1|Task_desc_1|Tag1_1|testSeg1|TEST_CTX1|ctx1Seg1
TASK_22APR_1|Code1_2|2|N|25/1/2014|25/11/2015|TASK_22apr_2|Task_desc_2|Tag1_2|testSeg1TEST_CTX1|ctx1Seg1
TASK_22APR_2|code2_1|3|N|25/12/2012|25/7/2015|TASK_22qpr_2_1|Task_desc_2|tag2_1|testSeg2|TEST_CTX2|ctx2Seg2
TASK_22APR_2|code2_2|3|Y|25/12/2012|25/7/2015|TASK_22qpr_2_2|Task_desc_2_2|tag2_2|testSeg2|TEST_CTX2|ctx2Seg2
cctx2Seg2
```

Set Enabled Lookups

To create a file containing the set enabled lookup types, include the following headers:

- **LookupType**: The lookup type.
- **Meaning**: The display name of the lookup type.
- **Description**: The description of the lookup type. This header is optional.
- **ModuleType**: The module with which the lookup type is associated.
- **ModuleKey**: The module code.
- **ReferenceGroupName**: Name of the reference group that contains the reference data set.

Here's a sample that contains two set enabled lookup types:

```csv
LookupType|Meaning|Description|ModuleType|ModuleKey|ReferenceGroupName
CODE_22APR_1|CODE_22apr_1|Code_desc_1|APPLICATION|FND|BU_APAC
CODE_22APR_2|CODE_22apr_2|Code_desc_2|APPLICATION|FND|BU_APAC
```

To create a file containing the set enabled lookup codes, include the following headers:

- **Required headers**:
  - **LookupType**: The lookup type.
  - **LookupCode**: The lookup code associated with the lookup type.
  - **DisplaySequence**: The sequence position at which the lookup code appears in the list of values.
  - **EnabledFlag**: Indicates the status of the lookup code, whether it's enabled for display or not.
  - **Meaning**: The display name of the lookup code.
  - **SetName**: Name of the reference data set.

- **Optional headers**:
  - **StartDateActive**: Beginning of the date range during which the lookup code is active and visible on the page. The format is dd/M/yyyy.
End Date Active: End of the date range during which the lookup code is active and visible on the page. The format is dd/M/yyyy.

Description: Description of the lookup code.

Tag: Any tag associated with the lookup code that may be used for a quick reference or retrieval of information.

Here's a sample file that contains the header values at the beginning and lists four set enabled lookup codes to be imported. For importing several lookup codes, add more entries in the same format.

<table>
<thead>
<tr>
<th>LookupType</th>
<th>LookupCode</th>
<th>DisplaySequence</th>
<th>EnabledFlag</th>
<th>StartDateActive</th>
<th>EndDateActive</th>
<th>Meaning</th>
<th>Description</th>
<th>Tag</th>
<th>SetName</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA_22APR_1</td>
<td>Code1_1</td>
<td>1</td>
<td>Y</td>
<td>25/12/2014</td>
<td>25/5/2015</td>
<td>DATA_22apr_1</td>
<td>Data_desc_1</td>
<td>Tag1_1</td>
<td>TEST SET CODE 2</td>
</tr>
<tr>
<td>DATA_22APR_1</td>
<td>Code1_2</td>
<td>2</td>
<td>N</td>
<td>25/1/2014</td>
<td>25/11/2015</td>
<td>DATA_22apr_2</td>
<td>Data_desc_2</td>
<td>Tag1_2</td>
<td>TEST SET CODE 3</td>
</tr>
<tr>
<td>DATA_22APR_2</td>
<td>Code2_1</td>
<td>3</td>
<td>N</td>
<td>25/12/2012</td>
<td>25/7/2015</td>
<td>DATA_22qpr_2_1</td>
<td>Data_desc_2</td>
<td>Tag2_1</td>
<td>TEST SET CODE 2</td>
</tr>
<tr>
<td>DATA_22APR_2</td>
<td>Code2_2</td>
<td>3</td>
<td>Y</td>
<td>25/12/2012</td>
<td>25/7/2015</td>
<td>DATA_22qpr_2_2</td>
<td>Data_desc_2_2</td>
<td>Tag2_2</td>
<td>TEST_ERR_CODE_Z</td>
</tr>
</tbody>
</table>

Related Topics
- Overview of Files for Import and Export
- Guidelines for File Import and Export
- Upload Files to WebCenter Content Server

FAQs for Define Lookups

How can I access predefined lookups?
Search for predefined lookups using any of the manage lookups tasks.

1. In the Setup and Maintenance work area, go to any of the following tasks that contains the lookups you're looking for:
   - Manage Standard Lookups
   - Manage Common Lookups
   - Manage Set-enabled Lookups

2. Enter any of the search parameters and click Search. If you don't know the lookup type or the meaning, use the Module field to filter search results.

3. Click a lookup type to view its lookup codes.

   Tip: Click the Query By Example icon to filter the lookup codes.

Related Topics
- Use Query By Example

How can I edit lookups?
On any of the Manage Lookups pages, you can edit the existing lookup codes of a lookup type or add new lookup codes. You can edit lookups using the following tasks in the Setup and Maintenance work area:

- Manage Standard Lookups
- Manage Common Lookups
- Manage Set-enabled Lookups
Each task contains a predefined set of lookup types that are classified and stored. Open a task to search and edit the required lookup. However, you may not be able to edit a lookup if its configuration level doesn't support editing.

**Why can't I see my lookup types?**

Lookup types are classified using tasks that involve a group of related lookups, such as Manage Geography Lookups. Each task gives you access only to certain lookup types. However, the generic tasks provide access to all lookups types of a kind, such as common lookups associated with the Manage Common Lookups task.

If the lookup types in an application are available in the standard, common, or set-enabled lookups view, they're are central to an application. However, lookup types defined for a specific application are managed using the task for that application.

**What's the difference between a lookup type and a value set?**

A **lookup type** consists of lookups that are static values in a list of values. Lookup code validation is a one to one match.

A table-validated **value set** may consist of values that are validated through a SQL statement, which allows the list of values to be dynamic. The following table brings out the differences between a lookup type and a value set.

<table>
<thead>
<tr>
<th>Area of Difference</th>
<th>Lookup Type</th>
<th>Value Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of values</td>
<td>Static</td>
<td>Dynamic if the list is table-validated</td>
</tr>
<tr>
<td>Validation of values</td>
<td>One to one match of meaning to code included in a lookup view, or through the determinant of a reference data set</td>
<td>Validation by format or inclusion in a table</td>
</tr>
<tr>
<td>Format type of values</td>
<td>char</td>
<td>varchar2, number, and so on</td>
</tr>
<tr>
<td>Length of value</td>
<td>Text string up to 30 characters</td>
<td>Any type of variable length from 1 to 4000</td>
</tr>
<tr>
<td>Duplication of values</td>
<td>Never. Values are unique.</td>
<td>Duplicate values allowed</td>
</tr>
<tr>
<td>Management</td>
<td>Both administrators and end-users manage these, except system lookups or predefined lookups at the system configuration level, which can't be modified.</td>
<td>Usually administrators maintain these, except some product flexfield codes, such as GL for Oracle Fusion General Ledger that the end-users maintain.</td>
</tr>
</tbody>
</table>

**Tip:** You can define a table-validated value set on any table, including the lookups table. Thus, you can change a lookup type into a table-validated value set that can be used in flexfields.

Both lookup types and value sets are used to create lists of values from which users select values.

A lookup type can't use a value from a value set. However, value sets can use standard, common, or **set-enabled** lookups.
What's a lookup tag used for?
A tag is an additional label attached to the lookup. Tags are user defined and can be grouped depending on the user's requirement to make search convenient and effective.

The same tag may be used across lookup categories. In such cases, tags are used as a refined search criterion to filter information across several groups and get the search result.

How can I search for a specific lookup code?
Use the Query By Example functionality to sort through hundreds of lookup codes for a lookup type, and display a specific lookup code. Enter the first few characters of the lookup code value in any of the relevant fields to filter the records.

Note: The search functionality is case sensitive.

Manage Messages

Messages
Messages provide users with information about business or application errors or warnings.

Typically, messages inform the users about the following:

- Missing or incorrect data
- Status of an application, page, or a business object
- Status of an ongoing process
- Result of a user action

Besides notifying users about the problem, messages provide guidance to users on taking corrective action. Messages also warn users about the consequences of a certain action.

Oracle provides a set of predefined messages that are stored in a message dictionary. You can create additional messages or modify the existing ones using the Manage Messages task in the Setup and Maintenance work area.

Note: Don't delete predefined messages unless you're sure that they aren't used anywhere.

Message Dictionary
The message dictionary stores messages that the application requires at run time. Messages are predefined for specific applications and modules, but a few are common messages that can be used in any application or module.

When you create messages, use the message text and the following components to cover additional details addressing users and help desk personnel:

- User Details: A detailed explanation of the message short text meant for users.
- Administrator Details: Details of the identified problem meant for the help desk personnel. The end users don't see this text.
- Cause: An end-user version of the cause of error.
• User Action: Instructions to users for addressing the identified problem. Where there is no guidance for end users, they must approach the help desk.
• Administrator Action: Corrective action that help desk personnel must take to correct the problem. This information isn't available to the end users.

Message Types
All messages must be associated with a message type. You can select the message type based on the message severity.

The available message types are:
  • Error
  • Warning
  • Information
  • UI String

Error Messages
Use the Error message to inform users about, for example, entering incorrect data or performing actions that trigger validation. Error messages also inform users how to correct the situation so that they can continue with their task.

For example: You can't specify a task without specifying the project.

Error messages also tell users about any serious problem with the application or process, and when they must seek assistance from the help desk. Some error messages trigger incidents or logs and have a mechanism to notify the help desk automatically.

Warning Messages
Use the Warning message type to inform users about an application condition or a situation that might require their decision before they can continue.

Warning messages:
  • Describe the reason for the warning and potential consequence of the selected or intended user action.
  • Can be either a question or a statement.

For example: You delete the primary user. Do you want to continue?

The message is usually followed by Yes and No buttons.

Information Messages
The Information message type tells users about changes in the application, a page, or a business object. These messages aren't triggered by users, and they don't have to take any immediate action in response.

For example: No events have been started or processed for this employee.

Use the Information message type to communicate information that's neither an error nor a warning.

UI String Messages
Use the UI string message type to store shorter messages such as UI prompts, titles, or translated text, in the message dictionary.
Message Groups

You can group messages by severity to internally define logging and classifying incident policies. You can group by category based on the functionality or program.

Category and severity values don't appear in logging entries, incidents, or on the UI.

**Note:** The values in both options are predefined lookups but you can modify them. However, the maximum size of this field is 30 characters.

To group the messages, in the Setup and Maintenance work area, use the Manage Messages task.

Group by Category

Use this option to group messages that relate to one functionality, such as a scheduled process, together into one category. Select one of the predefined categories to enable automatic incident creation when the error message activates. By default, the following categories are available:

- **Product:** Issues related to product functionality, setup, and maintenance. Such messages are typically intended for functional administrators or product super users.
- **System:** Issues concerning the application, database, technology stack, and so on. Such messages are typically intended for technical users such as application administrators or database administrators.
- **Security:** Issues concerning permissions, access, compliance, passwords, and so on. Such messages are typically intended for security administrators.

Group by Severity

This grouping attribute is very specific and indicates the severity of the message. You must set the severity to High to enable automatic incident creation for the message. The following are predefined values, but you can add more if required.

- **High:** Used for serious messages that completely stop the progress of an important business process or affect a large user community, and require help desk’s attention. Use this option to enable implicit incident creation for the message.
- **Medium:** Used for less severe and more isolated messages.
- **Low:** Used when you can't determine whether the message has a negative impact on end users or business processes.

Logging and Incidents

Select the **Logging Enabled** check box to include the UI message in the stored log file. To enable automatic incident creation when the error message appears on the UI, set the severity to High.

Incidents collect information about the application errors for which users may require assistance from help desk. An incident contains information about the state of the application at the time the problem occurred. Help desk can use the information in the incidents to resolve the problems.

Related Topics

- What’s an incident
- Run Diagnostic Tests
- Run Diagnostic Tests to Verify, Troubleshoot, and Analyze
Create and Edit Messages

You can create messages or edit the predefined messages stored in the message dictionary.

Creating a Message
To create a message, perform the following steps:

1. In the Setup and Maintenance work area, go to the Manage Messages task.
2. On the Manage Messages page, click the New icon.
3. On the Create Message page, enter details in each section.
4. In the Message Properties section:
   a. Enter a unique message name that helps you find the messages you create and avoid name conflicts with predefined messages. Use underscore as a separator if the name contains multiple parts.
   b. Select the application and module to associate the message with.
   c. Enter a unique number that can be used as an identifier for the message. Users can quote this number when they contact the help desk for assistance.
      Note: You can use any number between 10,000,000 and 10,999,999. This number range is allocated for the messages you create. At runtime, this number appears along with the application code after the message text, for example FND-2774.
   d. In the Translation Notes field, enter a description of the message indicating its use.
   e. Select the relevant message type, category, and severity.
   f. Select the Logging Enabled check box to create incidents or logs when messages appear on the UI.
5. In the Message Text section:
   a. In the Short Text field, provide the actual message text that appears on the page at runtime.
      The short text can include tokens that are placeholders for displaying dynamic values at runtime. However, to support easy translation, keep the message length (including values of tokens) within 160 characters in American English.
   b. In the User Details field, enter information for the users to know why the message appeared. You can also include information for the users to resolve the issue themselves.
      If your Short Text component has tokens that expand the text beyond the 160-character limit, move that portion of text here.
   c. In the Administrator Details field, provide a detailed technical explanation of the message. This field is only visible to the help desk.
   d. In the Cause field, provide a concise explanation of why the message appears. This text is visible to the users.
      This information is optional and is only applicable to messages of type Error and Warning. However, if you mention the cause, you must mention in the User Action field the action that users must take.
   e. In the User Action field, enter the user action to guide the users with steps to respond to the message and complete the task.
   f. In the Administrator Action field, provide information that the help desk can use to resolve the problem.
6. In the Message Tokens section, define tokens that you want to use in this message.
7. Click Save and Close.
Editing a Message
You can edit a predefined message or a message that you created.

To edit a message, search for a message on the Manage Messages page and perform the following steps:

1. Select the existing message and click the Edit icon.
2. On the Edit Message page, modify the existing details according to the instructions provided in the Creating a Message procedure.

Note: Don’t edit the message number for predefined messages.

3. Click Save and Close.

While creating and editing messages, you can translate the message details to the preferred languages without changing the language session of the application. To specify the translations in all the enabled language rows, use the Translation Editor option. Once the updates are made, users can view the translated text for the specific details.

Related Topics
• Enter or Edit Translated Text

Use of Tokens in Messages
Certain messages must display variable information at run time to help users clearly relate to the actual data and perform the required action. You can use tokens to contain variable values at run time, instead of writing a unique message for every possible situation.

Tokens are programmatic parts of message text that are placed within curly brackets when creating the message. Tokens serve as placeholders for the actual data. At run time, tokens dynamically display the actual text or value in the message, making a message specific to the situation. For example, the message "Enter an effective date that's the same as or later than {MATURITY_DATE}" contains the token {MATURITY_DATE}. At run time, instead of the token, the represented value (the actual date) appears. Thus, users see the message "Enter an effective date that's the same as or later than 25-APR-2015".

Use the Manage Messages task in the Setup and Maintenance work area to create and manage tokens. You must edit a message to define tokens for it. You can create tokens for a message and also delete them. However, you can’t edit or delete the predefined tokens.

Token Definition
To define a token, you must provide the following information:

• A unique name for the token.
• The type of data that the token replaces at run time. Available types are Date, Number, or Text.
• A description about what the token represents at run time.

Guidelines
Follow these general guidelines while defining tokens:

• Use curly brackets and all uppercase letters for the token names.
• Use underscore as a separator for a name containing two words or more.
• Don’t use a space between words.
The following table contains specific guidelines for each token data type.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Guideline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Use tokens for substituting any variable text element that qualifies as a noun.</td>
</tr>
<tr>
<td>Number</td>
<td>Plan carefully while using tokens for numbers especially, where a token could refer to either a singular or a plural number. You can use tokens for numbers representing an order, customer, or any other business object bearing a numeric value.</td>
</tr>
<tr>
<td>Date</td>
<td>Clearly define the context of the date, such as the start date, or end date, or a date range.</td>
</tr>
</tbody>
</table>

**Common Messages**

Message names that begin with FND_CMN are common messages. Each common message can appear in multiple places in any product family across Oracle Applications Cloud. For example, the FND_CMN_NEW_SRCH message can be used for any search to indicate that no results were found. Common messages of type error or warning are part of the message dictionary.

**Creating and Editing Common Messages**

You can create common messages for use in multiple places. However, ensure that you follow the predefined naming convention and numbering series associated with the application or module.

> **Note:** Don't use FND_CMN as the prefix for the messages you create because all the predefined common messages begin with it.

Common messages can be used in any application. Therefore, 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, ensure that the text is generic and applies to the entire site of Oracle Applications Cloud implementation.

**How can I make message components visible only to specific users?**

Use the Manage Administrator Profile Values task to determine the visibility of the message components. For the **Message Mode** profile option, set the profile value to either User or Administrator. Based on the set value, the administrator or user actions and details appear for the intended audience.

However, the message components are visible to the audience based on their predefined access rights. Anyone having a user level access can't view the Administrator message components. If you set the profile value to the administrators of a specific product, the message components are visible only to that specific audience.

> **Note:** If you don't set any value to the profile option, the visibility of the message component is determined by the default profile option settings.
Define Document Sequences

Document Sequences

You can assign a document sequence number to each business document or business event to uniquely identify it. For example, you can assign a document sequence number to an invoice that gets generated in response to a purchase order. However, you must enable the document sequencing option for that business document or event to start assigning the number. A document sequence number is useful in tracking completed or failed transactions.

You can set up document sequencing in three different modes:

- Automatic
- Manual
- Gapless

Note: Plan your document sequencing carefully before you start applying sequence numbers. Avoid switching to a different mode 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 automatically when the document is generated. That unique number is stored in the database. You can set an initial value for the numbering sequence. Thereafter, the numbering is sequential by date and time of creation. If you don't provide an initial value, the application sets the default initial value as 1.

Manual Sequencing

Use the manual sequencing mode to assign a unique number to each document before the document is generated. In manual sequencing, the numeric ordering and completeness of a transaction isn't automatically enforced. As a result, users can skip or omit numbers when entering the sequence value. However, each time a user assigns a number, 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. Sequence numbers aren't assigned to incomplete or failed documents. As a result, the sequence is maintained for all the successfully generated documents.

Additionally, you can control the gapless document sequencing by enforcing the Transaction Date Validation option. When enabled, this option checks for the transaction date of a particular document and assigns the sequence number accordingly, to chronologically maintain the documents. The sequence numbers and the transaction dates are chronologically correlated to prevent any mismatch of a new document sequence assigned to an older document or an older document sequence assigned to a new document.

Note: Use this type of sequencing only if necessary because it may affect the performance of the application and slow down transaction processing.

Related Topics
- Modules in Application Taxonomy
Document Sequence Categories

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.

Note: Once a document sequence category is created, you can’t 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’s available for use in 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.

Guidelines for Managing Document Sequences

Sequencing documents is an important business and legal requirement. Therefore, you must first decide the appropriate document sequence to use for a set of documents. Before you begin, here are a few prerequisites:

- Determine beforehand the mode of document sequencing, because you can’t switch to other types once a sequence is in use.
- Note details such as the document sequence and document sequence category, for later reference.
- Identify if there are any restrictions or configuration prerequisites.

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. The sequence definition never expires if you don’t provide an end date. Among the several options used in creating and editing document sequences, the following options are functionally more important and therefore must 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 must not be more than 11 digits. If you leave this field blank, the first document is automatically assigned a value of 1. Once a document sequence is defined, you can’t 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. 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.
- Select only those tables that belong to the application associated with the category.
- Once a category is defined, you can't switch to another 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, 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 don't 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 determinant type.

At run time, when users create documents, the document sequence to be assigned is determined based on the following:

- An active assignment that matches the correct combination of category
- The numbering method
- 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 can't 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.

Related Topics

- Guidelines for Managing Modules in Application Taxonomy

Define Trees

Overview of Trees

Trees are hierarchical data models that you can use to organize data, apply business rules, control data access, and improve performance while querying. For example, an application maintains data of an organization called Vision Corporation that has two departments: Marketing and Finance. The Finance department has two functional divisions:
Receivables and Payables. You can define a tree for Vision Corporation to establish a hierarchy across its departments, and their respective functional divisions. You can use the hierarchy to manage data at various levels of the organization.

To work with trees, in the Setup and Maintenance work area, use any of the following tasks:

- Manage Tree Structures: To create and update tree structures. You must first define a tree structure to create a tree.
- Manage Trees and Tree Versions: To create and update trees and their versions.
- Manage Tree Labels: To create and update tree labels.

**Tree Structures**
As the name suggests, tree structures provide you the framework to organize data such that you can establish a hierarchy for use by the tree. So, similar to a template, a tree structure guides the creation of a tree.

**Tree**
A tree is an instance of the tree structure. The root node is the highest nodal point of a tree. Child nodes branch off from the root node. Child nodes at the same level, branching off from a common parent node, are called siblings. Leaves are details branching off from a node but not extending further down the tree hierarchy. You can create trees for multiple data sources and share them across applications.

**Tree Versions**
A tree by default has only one version. If required, you can create and maintain more than one editable tree version. At any point, only one tree version must be active. If you edit an existing version, it changes from active to draft. To use it again, you must set it to active. Similar to any other version control system, versions of trees are maintained to track all the changes that a tree undergoes in its life cycle.

**Tree Labels**
Tree labels are short names given to trees and tree structures. You can label the tree versions for better accessibility and information retrieval. When nodes are created in a tree, the existing tree labels are automatically assigned to the new tree nodes. You can use any table to store the labels and register the label data source with the tree structure.

**Manage Tree Structures**

**Tree Structures**
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's 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.
Guidelines for Defining the Tree Structure

While creating a tree structure, you must specify important details on the Create Tree Structure: Specify Definition page. As the source of the tree structure, you may either select the predefined tree structures and proceed with the definition or create tree structures.

Tree Node Selection

The data in Tree Node table maps to the data in nodes of the tree structure. You must select the correct and most appropriate tree node table to define the tree structure, based on which you establish the tree hierarchy. This selection also affects the level of security that is set on a tree node and its child entities.

Tree Sharing Mode

Use the following options 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 is associated with a specific reference data set.

Modification

You can modify the predefined tree structures as well as those you create. However, modifying a predefined tree structure is restricted and permitted through additional privileges. Modification is limited to specific tree nodes and lower in the tree hierarchy.

Multiple Tree Versions

Although multiple tree versions can exist together, Oracle recommends only one version be active at any given time. However, if required, you can have more tree versions to be in the active state for the same date range. You can use this flexibility to select the tree version you want to implement.

Guidelines for Managing Tree Structures

You can create, edit, and delete tree structures. You can also change the status of a tree structure and audit the changes.

Creating and Editing Tree Structures

When you edit an active tree structure, the status of the tree structure and all associated trees and their versions changes to draft. To reuse a tree structure, create a copy of the tree without copying the associated trees and tree versions. After making changes, set the status again to active. If you delete a tree structure, all the associated trees and tree versions are automatically deleted.

For information about working with the offering-specific predefined tree structures, refer to the relevant product documentation.

Status

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

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

<table>
<thead>
<tr>
<th>Status</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft</td>
<td>In a modified state, or not yet in use.</td>
</tr>
<tr>
<td>Active</td>
<td>In use, indicating that one or more trees or tree versions are created from the tree structure.</td>
</tr>
</tbody>
</table>
Tree Structure Audit Results

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. To manually trigger an audit, select Audit from the Actions menu on the Manage Tree Structures page. The Tree Structure Audit Result table shows a list of validations that ran against the selected tree structure.

Audit Validators

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

<table>
<thead>
<tr>
<th>Validator</th>
<th>Page</th>
<th>Description (what's validated)</th>
<th>Possible Cause for Validation Failure</th>
<th>Suggested Corrective Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restrict By Set ID</td>
<td>Manage Tree Structures: Specify Data Sources</td>
<td>If you select the Reference Data Set check box for the Restrict Tree Node List of Values Based on option, each of its data source view objects must have a reference data set attribute.</td>
<td>Even when the check box is selected, one or more data source view objects doesn't contain a reference data set attribute.</td>
<td>If reference data set restriction is required for this tree structure, include a reference data set attribute on all data sources. Otherwise, deselect the check box.</td>
</tr>
<tr>
<td>Available Label Data Sources</td>
<td>Manage Tree Structures: Specify Data Sources</td>
<td>If you select a list item from Labeling Scheme to specify a labeling scheme, the label data source view object specified for each data source must be accessible. Also, the primary keys must be valid. This restriction doesn't apply if you select None from the list.</td>
<td>Any of the specified label data source view objects doesn't exist. Any of the specified label data source view objects doesn't have primary keys. When a label data source view object is initially defined,</td>
<td>Correct the specified label data source view object. Correct the primary keys of the specified label data source view object. Do one of the following: Correct the primary keys in</td>
</tr>
<tr>
<td>Validator</td>
<td>Page</td>
<td>Description (what's validated)</td>
<td>Possible Cause for Validation Failure</td>
<td>Suggested Corrective Action</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Row Flattened Table Name        | Manage Tree Structures: Specify Performance Options | You must specify a valid row flattened table for the tree structure. It can either be the standard row flattened table FND_TREE_NODE_RF or another table. | • The specified table doesn't exist in the database.  
• The specified table doesn't contain the same columns as the FND_TREE_NODE_RF table.                                                                                                                                   | Correct the row flattened table definition.                                                                                              |
| Available Data Sources          | Add Data Source                                    | Each data source view object specified for the tree structure must be accessible, and all its primary key attributes must be valid.                                                                                                         | • Any of the specified data source view objects doesn't exist.  
• When you define a data source view object, keep the Use non-defined primary key columns check box deselected. The database automatically registers the primary keys for the view object. Select this check box if you want the database to register the primary keys | • Correct the specified data source view object.  
• Correct the duplicate column in the registered primary keys.  
• Correct the primary keys of the specified data source view object.  
• Correct any mismatch in data types. |
<table>
<thead>
<tr>
<th>Validator</th>
<th>Page</th>
<th>Description (what's validated)</th>
<th>Possible Cause for Validation Failure</th>
<th>Suggested Corrective Action</th>
</tr>
</thead>
</table>
| Column Flattened Table Name      | Manage Tree Structures: Specify Performance Options | You must specify a valid column flattened table for the tree structure. It can either be the standard row flattened table FND_TREE_NODE_CF or another table. | • The specified table doesn’t exist in the database.  
• The specified table doesn’t contain the same columns as the FND_TREE_NODE_CF table. | Correct the column flattened table definition. |
| Restrict by Date                  | Manage Tree Structures: Specify Data Sources     | If you select the Date Range check box for 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 end date attributes. This validation doesn’t take place when | Even when the check box is selected, one or more of its data source view objects doesn’t contain effective start date and 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. |
### Guidelines for Specifying Data Sources for Tree Structures

The data sources provide the items for establishing hierarchy in a tree structure. In the tree management infrastructure, these data sources are Oracle 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 customers assignment. You may also choose not to have any labeling scheme. However, if you decide to use any of the labeling schemes, select the following additional options, to restrict the list of values that appear in 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 in 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 modified 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 Oracle ADF business components are used as data sources. To associate the view object with the tree structure, you can pick the code from Oracle 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.

### Guidelines for Improving the Performance of a Tree Structure

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**
- **BI 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 runtime 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 Object

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.
BI View Object
View objects from Business Intelligence can 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 BI view object generated by the tree management for business intelligence reporting and usage. The BI 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.

Search View Object
Specify the full name of the view object for the tree node to ensure that search operations performed on the tree node are efficient.

Manage Tree Labels

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

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

<table>
<thead>
<tr>
<th>Labeling Scheme</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>Labels that are automatically assigned based on the data source to which the tree node belongs. A level label points to a specific data source. For example, in a tree that reflects the organizational hierarchy of an enterprise, all division nodes appear on one level and all department nodes on another.</td>
</tr>
<tr>
<td>Group</td>
<td>Labels that you can arbitrarily assign to tree nodes.</td>
</tr>
<tr>
<td>Depth</td>
<td>Labels that are automatically assigned based on the depth of the tree node within the tree. No manual assignment is performed.</td>
</tr>
</tbody>
</table>

Note: In an unbalanced hierarchy, a level may not be equal to depth.

Manage Trees and Tree Versions

Guidelines for Managing Trees and Tree Versions
You can create and edit trees and tree versions depending upon the requirement. A tree can have one or more tree versions. 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. You can also select and copy the associated tree 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 modified 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. Each tree must contain a version.

Editing an existing tree provides you with the option of updating the existing version. You can also edit the existing version that lies nested in 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 canceled. To edit a tree version while keeping the status active select the Update tree nodes while keeping the tree version active Check Box.

Tree Version Audit Results
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. If you run an audit on a tree version, which is in Draft status, and want to change the status to Active after the audit is complete then select the Set tree version status to active after audit is successfully complete Check Box.

The Tree Version Audit Result table shows a list of validations that ran against the selected tree version.

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

<table>
<thead>
<tr>
<th>Validator</th>
<th>Description (what's checked)</th>
<th>Possible Cause for Validation Failure</th>
<th>Suggested Corrective Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effective Date</td>
<td>The effective start and end dates of the tree version must be valid.</td>
<td>The effective end date is set to a value that isn’t greater than the effective start date.</td>
<td>Modify the effective start and end dates such that the effective start date is earlier than the effective end date.</td>
</tr>
<tr>
<td>Validator</td>
<td>Description (what's checked)</td>
<td>Possible Cause for Validation Failure</td>
<td>Suggested Corrective Action</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Root Node</td>
<td>On the Manage Tree Structures: Specify Data Sources page, if the Allow Multiple Root Nodes check box for the Restrict Tree Node List of Values Based on option isn't selected, and if the tree structure isn't empty, the tree version must contain exactly one root node. This validation doesn't take place if the check box is selected.</td>
<td>Even if the check box is deselected, the tree version has multiple root nodes.</td>
<td>Modify the tree version such that there is exactly one root node.</td>
</tr>
<tr>
<td>Data Source Maximum Depth</td>
<td>For each data source in the tree structure, on the Data Source dialog box, if the data source is depth-limited, the data in the tree version must adhere to the specified depth limit. This validation doesn't apply to data sources for which the Maximum Depth field is set to Unlimited.</td>
<td>The tree version has data at a depth greater than the specified depth limit on one or more data sources.</td>
<td>Modify the tree version such that all nodes are at a depth that complies with the data source depth limit.</td>
</tr>
<tr>
<td>Duplicate Node</td>
<td>On the Data Source dialog box, if the Allow Duplicates check box isn't selected, the tree version must not contain more than one node with the same primary key from the data source. If the check box is selected, duplicate nodes are permitted.</td>
<td>Even when the check box is deselected, the tree version contains duplicate nodes.</td>
<td>Remove any duplicate nodes from the tree version.</td>
</tr>
<tr>
<td>Available Node</td>
<td>All nodes in the tree version must be valid and available in the underlying data source.</td>
<td>• A node in the tree version doesn't 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</td>
<td>Remove any orphaned nodes from the tree version. Update tree reference nodes so that they reference existing tree versions.</td>
</tr>
<tr>
<td>Validator</td>
<td>Description (what's checked)</td>
<td>Possible Cause for Validation Failure</td>
<td>Suggested Corrective Action</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Node Relationship</td>
<td>All nodes must adhere to the relationships mandated by the data sources registered in the tree structure.</td>
<td>The tree structure has data sources arranged in a parent-child relationship, but the nodes in the tree don't 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 must always be within Project nodes in the tree version. This validation fails if there are instances where a Project node is added as the child of a Task node.</td>
<td>Modify the tree version such that the nodes adhere to the same parent-child relationships as the data sources.</td>
</tr>
<tr>
<td>SetID Restricted Node</td>
<td>On the Manage Tree Structures: Specify Data sources page, if the Set ID check box is selected to enable the 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 doesn't apply when the check box isn't selected.</td>
<td>Even when the check box is selected, the tree version has nodes whose data source values belong to a different reference data set than the tree.</td>
<td>Modify the tree version such that all nodes in the tree have data sources with reference data set matching that of the tree.</td>
</tr>
<tr>
<td>Label Enabled Node</td>
<td>On the Manage Tree Structures: Specify Data Sources page, if a labeling scheme is specified for the tree structure by selecting a list item from the Labeling Scheme list, all nodes must have labels. This restriction doesn't apply when you select None from the Labeling Scheme list.</td>
<td>The tree structure has a labeling scheme but the tree version has nodes without labels.</td>
<td>Assign a label to any node that doesn't have a label.</td>
</tr>
<tr>
<td>Date Restricted Node</td>
<td>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</td>
<td>Even when the check box is selected, there are data source nodes that have a date range beyond the tree version's effective date range. For example, if the tree version is effective from Jan-01-2012 to Dec-31-2012, all nodes in the tree version must be effective from Jan-01-2012 to Dec-31-2012 at a minimum. It is acceptable for the nodes to be effective for a date range.</td>
<td>Ensure that all nodes in the tree version have effective date range for the effective date range for the tree version.</td>
</tr>
<tr>
<td>Validator</td>
<td>Description (what's checked)</td>
<td>Possible Cause for Validation Failure</td>
<td>Suggested Corrective Action</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>doesn't apply if the check box isn't selected.</td>
<td>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 isn't acceptable if the nodes are effective for none or only a part of the tree version's effective date range (for example, the node data source value are effective only from Jan-01-2012 to June-30-2012).</td>
<td></td>
</tr>
<tr>
<td>Multiple Active Tree Version</td>
<td>On the Manage Tree Structures: Specify Definition page, if the Allow Multiple Active Tree Versions check box isn't selected for the tree structure, there must not be more than one active tree version for a tree at any time. This restriction doesn't apply if the check box is selected.</td>
<td>Even when the check box isn't selected, there is more than one active tree version in the tree for the same date range.</td>
<td>Set no more than one tree version to Active within the same date range and set the others to inactive or draft status.</td>
</tr>
<tr>
<td>Range Based Node</td>
<td>On the Data Source dialog box, if the Allow Range Children check box isn't selected, range-based nodes aren't permitted from that data source. This restriction doesn't apply if the check box is selected.</td>
<td>Even when the check box isn't selected, there are range-based nodes from a data source.</td>
<td>Ensure that any range nodes in your tree version are from a data source that allows range children.</td>
</tr>
<tr>
<td>Terminal Node</td>
<td>On the Data Source dialog box, if the Allow Use as Leaves check box isn't selected, values from that data source can't be added as leaves (terminal nodes) to the tree version. This restriction doesn't apply if the check box is selected.</td>
<td>Even when the check box isn't selected, values from a data source are added as leaf nodes (terminal nodes).</td>
<td>Modify the tree version such that all terminal nodes are from data sources for which this check box is selected.</td>
</tr>
<tr>
<td>Usage Limit</td>
<td>On the Data Source dialog box, if the Use All Values option is selected to set the Usage Limit for the data source, every value in the data source must appear as a node in the tree. This restriction doesn't apply if None option is selected.</td>
<td>Even if the Use All Values option is selected, there are values in the data source that aren't in the tree version.</td>
<td>For each data source value that isn't yet available, add nodes to the tree version.</td>
</tr>
</tbody>
</table>
How Trees and Data Sources Work Together

Data sources are the foundation of tree management. 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 and Data Storage

Tree structures contain the metadata of the actual data and the core business rules that manifest in trees and tree versions. You can 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, the 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.

Tree Nodes

Tree nodes are points of data convergence where a tree branches into levels. Nodes are the building blocks of a tree structure and are attached to tree versions. Whenever you create or edit a tree version, you need to specify its tree node.

In the Setup and Maintenance work area, open the panel tab and click Search to search for the Manage Trees and Tree Versions task.

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

Usually, the nodes at a particular level represent similar information. For example, in a tree that reflects the organizational hierarchy, all nodes representing divisions appear at one level and all the department nodes on another. Similarly, in a tree that organizes a user’s product catalog, the nodes representing individual products might appear at one level and the nodes representing product lines on the immediate higher level.

The following node levels are in use:

- Root node: The highest node in the tree structure
- Parent node: The node that branches off into other nodes
- Child node: The node that's connected to a node higher in hierarchy (parent node)
- Sibling node: Nodes that are at the same level and belong to the same parent node
- Leaf node: Entities branching off from a node but not extending further down the tree hierarchy

Node Types

A tree node has the following node types.

- Single: Indicates that the node is a value by itself.
Define Profile Options

Overview of Profile Options

Profile options are a set of preferences that you use to centrally manage the user interface settings and application behavior.

You can use the profile options to manage, for example:

• User preferences to specify language or currency.
• Configuration choices to change the user interface skin or appearance of fonts.
• Processing options to determine how much of an activity needs to be logged and at which level.

In the Setup and Maintenance work area, use any of the following tasks:

• Manage Profile Options
• Manage Profile Categories
• Manage Administrator Profile Values

The following table contains a functional description of each task.

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Profile Options</td>
<td>Create new profile options or modify existing profile options, except some which are predefined and restricted to prevent any modifications.</td>
</tr>
<tr>
<td>Manage Profile Categories</td>
<td>Group the profile options based on their functional similarities.</td>
</tr>
<tr>
<td>Manage Administrator Profile Values</td>
<td>Set the profile values for the enabled profile options to control application behavior.</td>
</tr>
</tbody>
</table>

Hierarchy in Profile Levels

The hierarchy in profile levels determines the context for making a profile option effective.

You can enable a profile option at the following levels:

• Site level (lowest): The entire site of deployment
• User level (highest): A specific user
After you create or edit a profile option on the Manage Profile Options page, you must enable it. You can enable it at multiple levels. The setting at the highest enabled level takes precedence over the lower levels. User level is the highest in the hierarchy and always takes precedence over the settings at the site level.

On the Manage Administrative Profile Values page, set the profile value at any of the enabled levels of the profile option.

**Example of Profile Option Hierarchy**

The following table shows an example of setting the currency profile option at different levels.

<table>
<thead>
<tr>
<th>Profile Level</th>
<th>Hierarchy</th>
<th>Currency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site</td>
<td>Lowest</td>
<td>Euro</td>
</tr>
<tr>
<td>User</td>
<td>Highest</td>
<td>US Dollar</td>
</tr>
</tbody>
</table>

For this example, there are two users, John and Lisa. For John, the user-level profile value currency is set to US Dollar. If the Currency profile option is enabled only at the site level, both John and Lisa would see Euro as the default currency. If the profile option is enabled at the user level, users having a different currency set as their currency profile value would see only that currency. In this case, John would see US Dollar as the default currency. If the Currency profile option is enabled at the user level and there is no user level currency defined, the site level setting takes effect. When both site and user levels are enabled, the value for the user level takes precedence over the site level value.

**Set Profile Option Values**

Each profile option contains specific values that determine how it affects the application. You can add or modify the values for each profile option. Select or enter the value for one or more of the available levels (site, product, and user) so that each setting takes effect at the intended level.

**Setting the Profile Value**

1. In the Setup and Maintenance work area, go to the Manage Administrator Profile Values task.
2. On the Manage Administrator Profile Values page, search for and select the profile option.
3. In the Profile Values section, click Add. A new row is added for you to specify the following conditions:
   - **Profile Level**: Specify the level at which the profile value is to be set. If the profile value applies to the entire site, select Site.
   - **Product Name**: If you select Product as the profile level, select a product and specify the associated profile value.
   - **User Name**: If you select User as the profile level, select the user name and specify the associated profile value.
   - **Profile Value**: Select or enter the value corresponding to the selected profile level.

   **Note:** For an existing entry, you can modify only the profile value.

4. Repeat step 3 to add more rows and set the profile values.
5. Click Save and Close.
Note: Changes in the profile values take effect for a user on the next sign in.

**Import Profile Values**

Use the Import option on the Manage Administrator Profile Values page to import profile values in bulk and associate them with a profile option.

**Prerequisite**

The file containing the profile values is available in the document repository of Oracle WebCenter Content.

**Importing Profile Values**

To import profile values:

1. In the Setup and Maintenance work area, go to the Manage Administrator Profile Values task.
2. In the Profile Option: Profile Values section, from the Actions menu, select Import.
3. On the Import User Profile Values dialog box, select the WebCenter Content account to which the file was uploaded.
4. Enter the name of the file containing the profile values. The name here must match with the name of the file uploaded to the selected account.
5. Click Upload. The profile values are imported.

Note: If the import fails, click the link to the log file on the confirmation dialog box and examine the cause of failure.

**File Format for Importing Profile Values**

To import profile option values into the application, you create a text file with the values and upload the file to the Oracle WebCenter Content document repository. The file must follow a specific format, as described here. After the file is in the document repository, you can then import the profile values into the application following the instructions in the Importing Profile Option Values: Procedure topic.

To create a file containing the profile values, include the following headers:

- **ProfileOptionCode**: The profile option code.
- **LevelName**: Must contain the value (Site, Product, or User).
- **UserName**: Must correspond to the registered user name in the application. Don't provide any other shortened or coded name of the user.
- **ProfileOptionValue**: The profile value to be imported.

While creating the file, adhere to the following guidelines:

- Use a vertical bar or pipe ( | ) as a delimiter between fields for both header and value rows.
- Set the file encoding to UTF-8 without the Byte Order Mark (BOM), as per the Oracle WebCenter Content specification.

Here's a sample file that contains the header values at the beginning of the file, followed by line entries of the two profile values that are to be imported. For importing several profile values, add more line entries in a similar format.
Create and Edit Profile Options

Use profile options to manage user preferences and control the general function of applications. For example, you can control user preferences involving language, date, time, currency, and other similar general settings.

You can create a profile option and also determine the level at which that profile option takes effect. You can also define the profile values for the profile option. The profile values appear on the Manage Administrator Profile Values page when you select the profile option.

Creating a Profile Option

1. In the Setup and Maintenance work area, go to the Manage Profile Options task.
2. On the page, click Actions > New.
3. On the Create Profile Option page, fill all the fields with relevant details with specific attention to the following:
   a. Use the SQL Validation field to provide an SQL statement that displays the permissible profile values to be used. Using an SQL statement, you can select the values from another table and display them as a list of values.

   For example, to display the values Yes and No from a lookup table, you can use the following SQL statement:

   ```sql
   select MEANING, LOOKUP_CODE from FND_LOOKUPS where LOOKUP_TYPE='YES_NO'
   ```

   As a result, on the Manage Administrator Profile Values page, the profile values Yes and No are available for selection for that profile option.
   b. You can specify a date range to keep the profile option active during that period. Beyond the specified duration, the profile option automatically becomes inactive. If you no longer require the profile option, you must manually delete it from the Manage Profile Options page.

4. Click Save and Close.
5. On the Manage Profile Options page, search for the newly created profile option and from the results, select it.
6. In the Profile Option Levels section, do the following:
   a. In Enabled, select the levels at which you want to enable the profile option.

     ![Note](https://example.com/note.png)

     **Note:** You can enable a profile option at multiple levels, but a higher-level profile value overrides a lower-level value. Therefore, enable them only at the required levels.

   b. In Updatable, select the profile level at which you want implementors to have update privileges. Leave the check box deselected if you don't want the implementors to modify the profile values (they appear in read-only mode).

7. Click Save and Close.
To edit a profile option that you created, search for it and edit the necessary details.

Note: While creating and editing profile options and profile categories, you can translate the details to the preferred languages without changing the language session of the application. To specify the translations in all the enabled language rows, use the Translation Editor option. Once the updates are made, users can view the translated text for the specific details.

Related Topics
- Enter or Edit Translated Text

Profile Categories
You can create profile categories to group profile options based on their functional similarities and their use. In the Setup and Maintenance work area, use the Manage Profile Categories task.

Profile categories help administrators or implementors in retrieving profile options using a search criterion on the Manage Administrator Profile Values page.

Managing Profile Categories
Consider the following options while managing profile categories:
- Create profile categories and add existing profile options to them
- Add newly created profile options to existing user-defined profile categories

Note: While you can add a profile option to more than one category, some profile categories are predefined and restricted from any modifications. So, you can’t edit them or add profile options to them.

Setting Display Sequence for the Profile Options
You must set the display sequence for each profile option that you add to a profile category. Display sequence determines the order in which the profile options appear in a search result, based on the profile category. You can set the sequence beginning with zero or one for the first profile option to display, and proceed sequentially to assign the values to the remaining profile options.

The following table demonstrates the effect of the display sequence on the profile options when they're retrieved as search results.

<table>
<thead>
<tr>
<th>Profile Category</th>
<th>Included Profile Option - Assigned Display Sequence</th>
<th>Display Sequence of Profile Options in the Search Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attachments</td>
<td>• Attachment File Directory - 2</td>
<td>1. Indicate Attachments</td>
</tr>
<tr>
<td></td>
<td>• Indicate Attachments - 1</td>
<td>2. Attachment File Directory</td>
</tr>
</tbody>
</table>

How can I access predefined profile options?
Search for predefined profile options using the Manage Profile Options task.

1. In the Setup and Maintenance work area, go to the Manage Profile Options task.
2. On the page, enter any of the search parameters and click Search.
Tip: If you don’t know the profile option code or the display name, use the Application or Module fields to filter search results.

3. Click a profile option to view its details.

Define Flexfields

Overview of Flexfields

A flexfield is a set of placeholder fields associated with business objects and placed on the application pages to contain additional data. You can use flexfields to modify the business objects and meet enterprise data management requirements without changing the data model or performing any database programming. Flexfields help you to capture different data on the same database table and provide a means to modify the applications features.

For example, an airline manufacturer may require specific attributes for its orders that aren’t predefined. Using a flexfield for the order business object, you can create and configure the required attribute.

Types of Flexfields

Flexfields that you see on the application pages are predefined. However, you can configure the flexfields or modify their properties. Users see these flexfields as field or information attributes on the UI pages. To manage flexfields, use any of the following tasks in the Setup and Maintenance work area:

- **Manage Descriptive Flexfields**: Expand the forms on the application page to accommodate additional information that is important and unique to your business. You can use a descriptive flexfield to collect invoice details on a page displaying invoices.
- **Manage Extensible Flexfields**: Establish one-to-many data relationships and make application data context-sensitive. The flexfields appear only when the contextual data conditions are fulfilled. Thus, extensible flexfields provide more flexibility than the descriptive flexfields.
- **Manage Key Flexfields**: Store information combining several values, such as a number combination. The key flexfields represent objects such as accounting codes and asset categories.
- **Manage Value Sets**: Use a group of values to validate the data entered in the flexfields.

Note: You can manage value sets within the Manage Descriptive Flexfields or Manage Extensible Flexfields tasks.

Related Topics

- Modules in Application Taxonomy

Flexfield Components

A flexfield is made up of several data entities that store and render information pertaining to flexfield configuration. Flexfields are made up of the following components:

- Segments
Segments
A segment is a field within a flexfield and represents a single table column of your database. When configuring a flexfield, define the appearance and meaning of individual segments. Segments represent attributes of information. Segments can appear globally wherever the flexfield is implemented, or based on a structure or context. Each segment captures a single atomic value and represents an attribute of information.

The characteristics of a segment vary based on the type of flexfield in which it’s used.
- In key flexfields, a segment describes a characteristic of the entity. For example, a part number that contains details about the type, color, and size of an item.
- In a descriptive or extensible flexfield, a segment represents an information attribute on the application page. For example, details about a device containing components, some of which are global while the remaining are contextually dependent on the category of the device.

Value Sets
Users enter values into segments while using an application. A value set is a named group of values that validate the content of a flexfield segment. You configure a flexfield segment with a value set to enforce entries of only valid values for that segment.

The configuration involves the following tasks:
- Defining the values in a value set, including characteristics such as the length and format of the values.
- Specifying formatting rules or values from an application table or predefined list.

Multiple segments within a flexfield, or multiple flexfields, can share a single value set.

Contexts
Context-sensitive flexfield segments are available to an application based on a context value. You define contexts as part of configuring a flexfield. Users see global segments as well as any context-sensitive segments that apply to the selected context value.

In descriptive flexfields and extensible flexfields, you can reuse the context-sensitive segments that are based on the database columns, in multiple contexts.

Structures
Key flexfields have structures. Each key flexfield structure is a specific configuration of segments. Adding or removing segments, or rearranging their order, produces a different structure. You can reuse the segments that are based on the database columns, in multiple structures.

Note: You can translate all these flexfield components to the preferred languages without changing the language session of the application. To specify the translations in all the enabled language rows, use the Translation Editor option on the respective edit pages. Once the updates are made, users can view the translated text for the specific flexfield components at runtime.

Related Topics
- Enter or Edit Translated Text
Overview of Flexfield Configuration

Configuring a flexfield involves identifying the need for enhancing a business object with user-defined attributes and then integrating the attributes into deployment. In the case of key flexfields, configuring the flexfield involves identifying value set assignments and determining segment structures.

Overall Process for Configuring User-Defined Attributes

Before using flexfields to create attributes, familiarize yourself with the context layers and the configuration life cycle of the application. You can add attributes to a business object using a flexfield, if developers have registered that object to a flexfield. For descriptive and extensible flexfields, the overall configuration process involves the following:

1. Use the Highlight Flexfields feature from the Administration menu to find flexfields on pages associated with business objects.
2. Plan the flexfield configuration.
3. Plan flexfield validation.
4. Define the attributes by configuring the flexfield segments.
   a. Use the Manage Extensible Flexfields or Manage Descriptive Flexfields tasks, or use the Configure Flexfield icon button directly on the page where the flexfield is highlighted. For simple configurations, use the Add Segment, Add Context Value, and Edit Segment icon buttons directly on the page where the flexfield is highlighted.
   b. Optionally, validate the flexfield configuration.
   c. Optionally, deploy the flexfield to a sandbox for initial testing.
5. Deploy the flexfield to the mainline metadata to display the attributes on the application pages and to make them available for integration with other tools such as Oracle Business Intelligence.
6. Perform the necessary steps to integrate the attributes into the technology stack.

A simple configuration is limited to such actions as adding a format-only field or adding a field with a basic list of values.

Overall Process for Configuring User-Defined Keys

Using key flexfields, you can configure intelligent key codes containing meaningful parts according to your business practices. You configure the key flexfield to have one segment for each part that makes up your key code.

For key flexfields, the overall configuration process involves the following:

1. Use the Highlight Flexfields feature from the Administration menu to find flexfields on pages associated with business objects.
2. Plan the flexfield configuration.
3. Plan the flexfield validation.
4. Define the value sets before configuring the key flexfield segments by going to the Manage Value Sets task.
5. Define the key flexfield structures and their segments, and define structure instances for each structure.
   a. Use the Manage Key Flexfields task or the Configure Flexfield icon button directly on the page where the flexfield is highlighted.
   b. Optionally, validate the flexfield configuration.
   c. Optionally, deploy the flexfield to a sandbox for initial testing.
6. Deploy the flexfield to the mainline metadata to display it on the application pages and to make it available for integration with other tools such as Oracle Business Intelligence.
7. Perform the necessary steps to integrate the flexfield into the technology stack.
Related Topics

- Overview of Context Layers
- Overview of Configuration Life Cycle

Flexfields at Runtime

Business objects have an associated descriptive or extensible flexfield. Using these, you can create attributes for the business object at run time. Some business objects have an associated key flexfield for configuring flexible multiple part keys.

Finding Flexfields on a Page

At run time, the attributes you define as flexfield segments appear in the application page just like any other attribute. However, each type of flexfield appears in a different way.

The following characteristics help you determine the type of flexfield on the application page:

- Descriptive flexfield segments appear as label and field pairs or as a table of fields that correspond to the column headers. The fields represent the flexfield segments and accept values that derive from 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.
- Key flexfields appear in the application page as a field with a key flexfield icon, where the field's value is a collection of segments.

To locate flexfields on a page, in the global header, select your user name and in the Settings and Actions menu, select **Highlight Flexfields**. The page renders in a special mode, displaying the location of flexfields, if any, on the page. Do the following:

- Hover over the **Information** icon to view flexfield details.
- Click the **Configure Flexfield** icon to manage the flexfield using the Manage Flexfields task.
- Click the **Add Context Value**, **Add Segment**, or **Edit Segment** icons to add a context value or edit a global or context-sensitive flexfield segment. This applies to both descriptive and extensible flexfields.

**Note:** You can't create attributes on all flexfields. For example, some flexfields are protected, and you either can't edit their configurations at all, or can do only limited changes to them. Consult the product-specific documentation to verify whether there are any restrictions on using the flexfield.

All segments of a single flexfield are grouped together by default. The layout and positions of the flexfield segments depend on where the application developer places the flexfield on the page. Flexfields may also be presented in a separate section of the page, in a table, or on their own page or a dialog box. You can use Oracle Composer to edit the layout, position, or other display features of the flexfield segments.

When you no longer want to view the flexfields on a page, select **Unhighlight Flexfields** from the Administration menu.
Flexfield Modification Using Page Composer

You can use Page Composer to modify flexfields specific to a page.

**Extensible Flexfield Modification**

In Page Composer, open the page with the flexfield you want to modify. Switch to Source view, and look for a region that is bound to an EffContextsPageContainer task flow. This is the container for the extensible flexfield attributes and contexts. To view the flexfield code and identifying information, open the properties panel for the region. To modify any component within the region, select the desired tag and click Edit.

**Descriptive Flexfield Modification**

In Page Composer, open the page with the flexfield you want to modify. Switch to Source view, and look for the `<descriptiveFlexfield>` element of that flexfield. Open the properties panel for this element to view the flexfield code and identifying information. Within the properties panel, you may modify properties for the global and context-sensitive segments or re-order the segments on the page.

>Note: Flexfield segments can't be edited using their individual component elements. They can only be edited from the properties panel of the `<descriptiveFlexfield>` element they belong to. To find these components, open the properties panel of the descriptive flexfield, switch to the relevant tab, and search using the unique identifying information. For instance, if you need to find a context sensitive segment, open the properties panel for the flexfield, go to the Flexfield Context Segments tab, and search for the segment using its Context Value and Segment Code.

**Import Extensible Flexfields**

Use the Import option on the Manage Extensible Flexfield page to bulk import the flexfield data that includes value set, context, and context segment details.

**Prerequisite**

The files containing the flexfield details are available in the document repository of Oracle WebCenter Content.

**Importing Flexfields**

To import flexfields:

1. Sign in to the application as an implementation consultant or an administrator.
2. In the Setup and Maintenance work area, open the Manage Extensible Flexfield task or a similar task for importing flexfields.
3. In Search Results, from the Actions menu, select Import.
4. On the Upload Flexfield Data dialog box, select the WebCenter Content account to which the files were uploaded.
5. Enter the names of the separate files containing the value set, context, and context segment information. The names here must match with the names of the files uploaded to the selected account.
6. Click Upload. The flexfield details are imported.

>Note: If the import fails, click the link to the log file on the confirmation dialog box and examine the cause of failure.
File Format for Importing Extensible Flexfields

To import extensible flexfields into an application, you create separate text files for the value sets, contexts, and context segments. Then, you upload them to the Oracle WebCenter Content document repository. Each file must follow a specific format, as described here. After the files are placed in the document repository, you can import the values sets, contexts, and context segments into the application.

While creating the file, adhere to the following guidelines:

- Use a vertical bar or pipe ( | ) as a delimiter between fields for both header and value rows.
- Set the file encoding to UTF-8 without the Byte Order Mark (BOM) as per the Oracle WebCenter Content specification.

The following sections contain specific details about each file format.

Prerequisite

You must have worked with flexfields in Oracle Cloud applications.

Value Sets

To create a file containing value sets, include the headers listed in the following table:

<table>
<thead>
<tr>
<th>Header</th>
<th>Data Type</th>
<th>Data Size</th>
<th>Description</th>
<th>Required or Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueSetCode</td>
<td>String</td>
<td>60</td>
<td>This value identifies your value set across components.</td>
<td>Required</td>
</tr>
<tr>
<td>ModuleType</td>
<td>String</td>
<td>60</td>
<td>This value determines the module type of your value set.</td>
<td>Required</td>
</tr>
<tr>
<td>ModuleKey</td>
<td>String</td>
<td>60</td>
<td>This value determines the specific module of your value set.</td>
<td>Required</td>
</tr>
<tr>
<td>ValidationType</td>
<td>String</td>
<td>30</td>
<td>This value determines your value set type. The values you can use are DEP, INDEP, FORMAT, SUBSET, and RELATED.</td>
<td>Required</td>
</tr>
<tr>
<td>ValueDataType</td>
<td>String</td>
<td>30</td>
<td>This value determines the data type that your value set uses. Supported data types are VARCHAR2, NUMBER, DATE, and TIMESTAMP.</td>
<td>Required</td>
</tr>
<tr>
<td>Header</td>
<td>Data Type</td>
<td>Data Size</td>
<td>Description</td>
<td>Required or Optional</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------</td>
<td>-----------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>ValueSubType</td>
<td>String</td>
<td>30</td>
<td>This value determines the data subtype your value set uses. Supported data subtypes are TEXT, TIME_HM, NUMERIC, TIME_HMS, and TRANSLATED</td>
<td>Required for ValueDataType VARCHAR2</td>
</tr>
<tr>
<td>MaximumLength</td>
<td>Integer</td>
<td>-</td>
<td>This value determines the maximum length of values in your value set.</td>
<td>Required for ValueDataType VARCHAR2</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>240</td>
<td>This value gives your value set a description.</td>
<td>Optional</td>
</tr>
<tr>
<td>Precision</td>
<td>Integer</td>
<td>-</td>
<td>This value determines the number of digits in the number data you add in your value set.</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Use this header only for ValueDataType NUMBER</td>
<td></td>
</tr>
<tr>
<td>Scale</td>
<td>Integer</td>
<td>-</td>
<td>This value determines the number of digits after the decimal point in the number data you add in your value set.</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Use this header only for ValueDataType NUMBER</td>
<td></td>
</tr>
<tr>
<td>UppercaseOnlyFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, then your value set will accept only uppercase characters. If it's N, then both uppercase and lowercase characters are supported.</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Use this header only for ValueDataType VARCHAR2</td>
<td></td>
</tr>
<tr>
<td>ZeroFillFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, then zeroes are added to the left of the text you add in your value set, till the text length matches your MaximumLength. If the value is N, no zeroes are added.</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Use this header only for ValueDataType VARCHAR2</td>
<td></td>
</tr>
<tr>
<td>SecurityEnabledFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, you can add a data security resource to</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Header</td>
<td>Data Type</td>
<td>Data Size</td>
<td>Description</td>
<td>Required or Optional</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>your value set. If the value is N, no data security resource is added.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DataSecurityObjectName</td>
<td>String</td>
<td>60</td>
<td>This value selects the data security resource you want to add to your value set. This value must match a data security resource in the application.</td>
<td>Required if SecurityEnabledFlag is set to Y.</td>
</tr>
<tr>
<td>MinimumValue</td>
<td>String</td>
<td>150</td>
<td>This value specifies the minimum value for the values in your value set.</td>
<td>Optional</td>
</tr>
<tr>
<td>MaximumValue</td>
<td>String</td>
<td>150</td>
<td>This value specifies the maximum value for the values in your value set.</td>
<td>Optional</td>
</tr>
<tr>
<td>IndependentValueSetCode</td>
<td>String</td>
<td>60</td>
<td>This value determines the independent value set your dependent or subset value set is linked to.</td>
<td>Required for ValidationType DEP and SUBSET. Don't use this header for validation types other than DEP and SUBSET.</td>
</tr>
</tbody>
</table>

Here's a sample file that contains the header values at the beginning of the file, followed by line entries of the three value sets that are to be imported. For importing several value sets, add more line entries in a similar format.

```
ValueSetCode|ModuleType|ModuleKey|ValidationType|ValueDataType|ValueSubtype|MaximumLength|Description
VS_TEST_91|APPLICATION|FND|INDEP|VARCHAR2|TEXT|2|desc1
VS_TEST_92|APPLICATION|FND|INDEP|VARCHAR2|TEXT|3|desc2
VS_TEST_93|APPLICATION|FND|INDEP|VARCHAR2|TEXT|3|desc3
```

**Context**

To create a file containing the contexts, include the headers in the following table:

<table>
<thead>
<tr>
<th>Header</th>
<th>Data Type</th>
<th>Data Size</th>
<th>Description</th>
<th>Required or Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplicationId</td>
<td>Long</td>
<td>-</td>
<td>This value determines the Application to which your flexfield and context belongs to.</td>
<td>Required</td>
</tr>
<tr>
<td>Header</td>
<td>Data Type</td>
<td>Data Size</td>
<td>Description</td>
<td>Required or Optional</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>EFFCode</td>
<td>String</td>
<td>40</td>
<td>This value is the code of the extensible flexfield to which you're adding your context. This value should match the code of an extensible flexfield that's already in the application.</td>
<td>Required</td>
</tr>
<tr>
<td>ContextCode</td>
<td>String</td>
<td>80</td>
<td>This value is the code for your context.</td>
<td>Required</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>80</td>
<td>This value determines the display name of your context.</td>
<td>Required</td>
</tr>
<tr>
<td>MultirowFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, then your context will support multi-row behavior. If the value is N, only single row behavior is supported.</td>
<td>Required</td>
</tr>
<tr>
<td>EnabledFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, then your context is enabled. If the value is N, then your context is disabled.</td>
<td>Required</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>240</td>
<td>This value gives your context a description.</td>
<td>Optional</td>
</tr>
<tr>
<td>TranslatableFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, then the segments in your context are translatable. If the value is N, they're not.</td>
<td>Optional</td>
</tr>
<tr>
<td>ContextIdentifier</td>
<td>String</td>
<td>30</td>
<td>This value determines the API name for your context. There are a set of conventions to be followed when naming APIs.</td>
<td>Optional</td>
</tr>
<tr>
<td>InstructionHelpText</td>
<td>String</td>
<td>400</td>
<td>This text gives the user instructions about how to use the context region.</td>
<td>Optional</td>
</tr>
<tr>
<td>Header</td>
<td>Data Type</td>
<td>Data Size</td>
<td>Description</td>
<td>Required or Optional</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>FlexfieldUsageCode</td>
<td>String</td>
<td>30</td>
<td>This value determines the usage code for your context. The value should match a usage code in the application.</td>
<td>Required</td>
</tr>
<tr>
<td>ViewPrivilegeName</td>
<td>String</td>
<td>400</td>
<td>This value determines which privileges can view this context usage. The value should match a privilege in the application.</td>
<td>Optional</td>
</tr>
<tr>
<td>EditPrivilegeName</td>
<td>String</td>
<td>400</td>
<td>This value determines which privileges can edit this context usage. The value should match a privilege in the application.</td>
<td>Optional</td>
</tr>
<tr>
<td>CExtAttribute1</td>
<td>String</td>
<td>150</td>
<td>This header adds a user defined attribute to a context usage. You can use up to 5 headers of this type, CExtAttribute1 to CExtAttribute5 to add your user defined attributes.</td>
<td>Optional</td>
</tr>
<tr>
<td>CONTEXT_CExtAttribute1</td>
<td>String</td>
<td>150</td>
<td>This header adds a user defined attribute to a context. You can use up to 5 headers of this type, CONTEXT_CExtAttribute1 to CONTEXT_CExtAttribute5 to add your user defined attributes.</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Here's a sample file that contains the header values at the beginning and lists three contexts to be imported. For importing several contexts, add more entries in the same format.

```
ApplicationId|EFFCode|ContextCode|Name|EnabledFlag|MultirowFlag|Description|FlexfieldUsageCode|ViewPrivilegeName|CExtAttribute1|CExtAttribute2|CExtAttribute3|CExtAttribute4|CExtAttribute5
0|FLEX_SN_EFF1|OBJ_TEST_4|Object test 44|Y|N|desc 44 3363|FLEX_SN_EFF1_USAGE2|flex_sn_sitems_view|||
0|FLEX_SN_EFF1|OBJ_TEST_3|Object test 33|Y|N||FLEX_SN_EFF1_USAGE1|flex_sn_view|||NEW_TEST11
0|FLEX_SN_EFF1|OBJ_TEST_3|Object test 33|Y|N|new desc aug 14|FLEX_SN_EFF1_USAGE2|flex_sn_sitems_edit|CE1_TESTupd_aug15|||
```
## Context Segment

To create a file containing context segments, include the headers in the following table:

<table>
<thead>
<tr>
<th>Header</th>
<th>Data Type</th>
<th>Data Size</th>
<th>Description</th>
<th>Required or Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplicationId</td>
<td>Long</td>
<td>-</td>
<td>This value determines the Application to which your flexfield, context, and segment belongs to.</td>
<td>Required</td>
</tr>
<tr>
<td>EFFCode</td>
<td>String</td>
<td>40</td>
<td>This value is the code of the extensible flexfield to which you’re adding your segment. This value should match the code of an extensible flexfield that’s already in the application.</td>
<td>Required</td>
</tr>
<tr>
<td>ContextCode</td>
<td>String</td>
<td>80</td>
<td>This value is the code for the context to which you’re adding your segment. This value should match a context code that’s in the application.</td>
<td>Required</td>
</tr>
<tr>
<td>SegmentCode</td>
<td>String</td>
<td>30</td>
<td>This value is the code for your segment.</td>
<td>Required</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>60</td>
<td>This is the back end name for your segment.</td>
<td>Required</td>
</tr>
<tr>
<td>ColumnName</td>
<td>String</td>
<td>30</td>
<td>This value determines the table column your segment uses to save data. This value must match a column that’s in the application.</td>
<td>Required</td>
</tr>
<tr>
<td>ValueSetCode</td>
<td>String</td>
<td>60</td>
<td>This value is the code for the value set you want to use in your segment. This value must match a value set code in the application.</td>
<td>Required</td>
</tr>
<tr>
<td>Header</td>
<td>Data Type</td>
<td>Data Size</td>
<td>Description</td>
<td>Required or Optional</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>DisplayType</td>
<td>String</td>
<td>30</td>
<td>This value determines the display type of the segment.</td>
<td>Required</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid values for this attribute are TEXT_BOX, TEXT_AREA, RICH_TEXT_EDITOR,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HIDDEN, LOV, POP_UP_LIST, DROP_DOWN_LIST, RADIO_BUTTON_GROUP, STATIC_URL,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CHECKBOX, COLOR, DATE_TIME.</td>
<td></td>
</tr>
<tr>
<td>Prompt</td>
<td>String</td>
<td>80</td>
<td>This value determines the display name of the segment.</td>
<td>Required</td>
</tr>
<tr>
<td>ShortPrompt</td>
<td>String</td>
<td>80</td>
<td>This value determines the shortened display name of the segment.</td>
<td>Required</td>
</tr>
<tr>
<td>EnabledFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, then your segment is enabled. If the value is</td>
<td>Required</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>N, then it's disabled.</td>
<td></td>
</tr>
<tr>
<td>RequiredFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, user input for this segment is mandatory during run time. If the value is N, user input is optional.</td>
<td>Required</td>
</tr>
<tr>
<td>ReadOnlyFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, then your segment is read-only. If the value is N, it's not.</td>
<td>Required</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>240</td>
<td>This value gives your segment a description.</td>
<td>Optional</td>
</tr>
<tr>
<td>UOMclass</td>
<td>String</td>
<td>20</td>
<td>This value determines the unit for the data you add in your segment.</td>
<td>Optional</td>
</tr>
<tr>
<td>Header</td>
<td>Data Type</td>
<td>Data Size</td>
<td>Description</td>
<td>Required or Optional</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>TerminologyHelpText</td>
<td>String</td>
<td>80</td>
<td>This text provides a description for the segment.</td>
<td>Optional</td>
</tr>
<tr>
<td>InFieldHelpText</td>
<td>String</td>
<td>160</td>
<td>This text provides instructions on how to use the segment.</td>
<td>Optional</td>
</tr>
<tr>
<td>SequenceNumber</td>
<td>Integer</td>
<td>-</td>
<td>This value determines the order in which your segments are displayed in your context.</td>
<td>Optional</td>
</tr>
<tr>
<td>DefaultType</td>
<td>String</td>
<td>30</td>
<td>This value determines the type of default value your segment takes if the user doesn’t enter any data in your segment. The types are CONSTANT, SQL, and GROOVY_EXPRESSION</td>
<td>Optional</td>
</tr>
<tr>
<td>DefaultValue</td>
<td>String</td>
<td>4000</td>
<td>This value determines the default value your segment takes if the user doesn’t enter any data.</td>
<td>Optional</td>
</tr>
<tr>
<td>DisplayWidth</td>
<td>Integer</td>
<td>-</td>
<td>This value determines the display width of your segment, in terms of maximum number of characters displayed in a line.</td>
<td>Optional</td>
</tr>
<tr>
<td>DisplayHeight</td>
<td>Integer</td>
<td>-</td>
<td>This value determines the display height of your segment, in terms of maximum number of lines displayed in the segment.</td>
<td>Optional</td>
</tr>
<tr>
<td>CheckboxCheckedValue</td>
<td>String</td>
<td>30</td>
<td>This value determines the value of a checked check box in your segment.</td>
<td>Required for display type CHECKBOX.</td>
</tr>
<tr>
<td>CheckboxUncheckedValue</td>
<td>String</td>
<td>30</td>
<td>This value determines the value of an unchecked check box in your segment.</td>
<td>Required for display type CHECKBOX.</td>
</tr>
<tr>
<td>Header</td>
<td>Data Type</td>
<td>Data Size</td>
<td>Description</td>
<td>Required or Optional</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>RangeType</td>
<td>String</td>
<td>30</td>
<td>This value sets two fields as two ends of a range of values. Set value as LOW or HIGH for your low end and high end fields respectively.</td>
<td>Optional</td>
</tr>
<tr>
<td>BIEnabledFlag</td>
<td>String</td>
<td>1</td>
<td>This value determines whether your segment is BI enabled. Y means it’s, N means it’s not.</td>
<td>Optional</td>
</tr>
<tr>
<td>MultirowUniqueKeyFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, the segment is marked as the unique key for a multirow context. Default value for this header is N. At least 1 segment in a multirow context must be marked as the unique key.</td>
<td>Required for multi-row contexts</td>
</tr>
<tr>
<td>ShowValueDescription</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, the value set description for the value set associated to the segment is displayed. If the value is N, it’s not displayed.</td>
<td>Optional</td>
</tr>
<tr>
<td>SegmentIdentifier</td>
<td>String</td>
<td>30</td>
<td>This value determines the API name for your segment. There are a set of conventions to be followed when naming APIs.</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Here’s a sample file that contains the header values at the beginning and lists a context segment to be imported. For importing several context segments, add more entries in the same format.

```
<table>
<thead>
<tr>
<th>ApplicationId</th>
<th>EFFCode</th>
<th>ContextCode</th>
<th>SegmentCode</th>
<th>Name</th>
<th>ColumnName</th>
<th>ValueSetCode</th>
<th>DisplayType</th>
<th>Prompt</th>
<th>ShortPrompt</th>
<th>EnabledFlag</th>
<th>RequiredFlag</th>
<th>ReadOnlyFlag</th>
<th>Description</th>
<th>UomClass</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>FLEX_SN_EFF1</td>
<td>ColdSourceTargetContext</td>
<td>Cold_TEST_SEG1</td>
<td>cold seg 1</td>
<td>ATTRIBUTE_CHAR3</td>
<td>EFF_BASIC_FMT_CHR</td>
<td>TEXT_BOX</td>
<td>Prompt</td>
<td>Short Prompt</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>TEST desc aug 14_3PM</td>
<td></td>
</tr>
</tbody>
</table>
```

**Related Topics**

- Overview of Files for Import and Export
- Upload Files to WebCenter Content Server
How Flexfields Work with Oracle Application Cloud Architecture

To capture additional data, administrators or implementors configure flexfield segments that represent attributes of business objects. Business objects are enabled for both descriptive flexfields and extensible flexfields.

The following figure shows the layers involved in configuring a flexfield:

- The business entity table and metadata in the database.
- The ADF business component objects. These are derived from the metadata and stored in Oracle Metadata Services (MDS) repository.
- The user interface where fields defined by the flexfield segments are rendered.
The following figure illustrates that 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's available for configuration. Administrators and implementation consultants configure segments and other properties of the available flexfields. This information is stored as additional flexfield metadata in the database. Deploying the flexfield generates ADF business components based on the flexfield metadata in the database.

The following aspects are important in understanding how flexfields and Oracle Applications Cloud architecture work together:

- Integration
- Deployment
• Import and export
• Run time
• Patching

Integration
The attributes that you add by configuring flexfields are available throughout the Oracle Fusion Middleware technology stack. You can use the flexfield segment’s Application Programming Interface (API) to identify segments and integrate the flexfields in the following:

• User interface pages
• Service-oriented Architecture (SOA) infrastructure
• Oracle Business Intelligence
• Extended Spreadsheet Database (ESSbase)

Flexfield configurations are preserved across application updates.

Deployment
The metadata for the flexfield is stored in the application database as soon as you save your configuration changes. Deploying the flexfield generates the ADF business components so that the run time user interface reflects the latest flexfield definition in the metadata.

Importing and Exporting
Using the Setup and Maintenance work area, you can import and export flexfields across the implementation site. The deployment status must be either Deployed or Deployed to sandbox. Therefore, before you attempt migration, verify and ensure that a flexfield is successfully deployed.

Run Time
The latest definitions of a flexfield reflect on the user interface at run time only if the flexfield is deployed. When the user interface accesses a business object, the deployed flexfield definition identifies the attributes associated with the captured values. On a page, if you add display configurations for a flexfield using Oracle Composer, the same flexfield segments can appear differently on different pages.

Patching
Flexfield configurations are preserved during patching and upgrading.

Flexfield Management

Considerations for Managing Flexfields
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 users. Optionally, you can modify the UI page to change how the flexfield segments appear to users on that page.
The following figure shows the processes involved in making flexfields available to users. The tasks in the Define Flexfields activity let administrators configure and deploy flexfields. After you configure and deploy a flexfield to a sandbox, deploy it again to the mainline metadata so that it's available to the users.

Consider the following aspects of managing flexfields:

- Registering flexfields
- Planning flexfields
- Configuring flexfields
- Enabling a flexfields segment for business intelligence
- Deploying flexfields
- Optionally changing a flexfield segment's appearance in a user interface page
Registering Flexfields

A flexfield must be registered before it can be configured. Therefore, application development registers flexfields so that they are available to administrators and implementation consultants for configuration. The registration involves reserving columns of entity tables for use in flexfields. For more information about registering flexfields, see Oracle Fusion Applications Developer’s Guide.

Planning Flexfields

Before you begin planning flexfields, determine what type is appropriate to your needs, and which business objects are available for modifying flexfields. All flexfields consist of segments which represent attributes of an entity. The value a user enters for an attribute is stored in a column of the entity table. Carefully plan flexfields before configuring them. Before configuring new segments for your flexfields, be sure to plan their implementation carefully.

If you have determined that a business object supports flexfields, and those flexfields have been registered, you can begin planning their configuration. Note the code name of the flexfield you intend to configure so that you can find it easily in the Define Flexfield activity. In some cases you can determine and configure how the flexfield appears on the page. See Oracle Applications Cloud Help for specific products to determine any restrictions on using product-specific flexfields.

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:

- Use the manage flexfield tasks in the Setup and Maintenance work area.
- Use the Highlight Flexfields command in the Administration menu while viewing a run time page.
  - Use the **Configure Flexfield** icon button to manage all aspects of a flexfield, such as change a segment’s sequence number or configure a flexfield segment’s business intelligence label.
  - Use the **Add Segment** and **Edit Segment** icon buttons to add and edit descriptive or extensible flexfield segments with simple configurations.
  - Use the **Add Context** icon button to add descriptive or extensible flexfield context values.

Configuring a flexfield includes the following:

- Defining *value sets* against which the values entered by users are validated
- Defining the structure or *context* of the segments in the flexfield
- Specifying the identifying information for each segment
- Specifying the display properties such as prompt, length and data type of each flexfield segment
- Specifying valid values for each segment, and the meaning of each value within the application

**Tip:** You can create value sets while creating descriptive and extensible flexfield segments. However, define value sets before configuring key flexfield segments that use them, because you assign existing value sets while configuring key flexfield segments.

When creating table-validated, independent, dependent, or subset value sets while creating descriptive and extensible flexfield segments, you can optionally specify to display the description of the selected value next to the segment at run time. You can assign sequence order numbers to global segments and to context-sensitive segments in each context. Segment display is always in a fixed order based on the segments’ sequence numbers. You cannot enter a number for one segment that is already in use for a different segment. Therefore, you may consider numbering the segments in multiples, such as 4, 5, or 10, to make it easy to insert new attributes.
A flexfield column is assigned to a new segment automatically, but you can change the assignment before saving the segment. If you must set a specific column assignment for a segment, create that segment first to ensure that the intended column isn't automatically assigned to a different segment.

Enabling a Flexfield Segment for Business Intelligence
You can enable flexfield segments for business intelligence if the flexfield is registered in the database as an Oracle Business Intelligence-enabled flexfield. For more information about enabling segments for business intelligence, see points to consider when enabling descriptive, extensible, and key flexfield segments for business intelligence. For extensible flexfield segments, you can't assign labels to equalize segments across contexts that are semantically equivalent.

Deploying Flexfields
Once you have configured a flexfield, you must deploy it to make the latest definition available to run time users. In the Define Flexfields tasks, you can deploy a flexfield using either of the following commands:

- The Deploy Flexfield command deploys a flexfield to the mainline metadata. This command is for general use in a test or production environment.
- The Deploy to Sandbox command deploys a flexfield to sandbox. This command is for confirming that the flexfield is correctly configured before deploying it to the mainline metadata.

In Highlight Flexfields mode, when using the:

- **Add Context, Add Segment, and Edit Segment** tools for extensible flexfields, use the Save command to save your changes. Then use the Deploy command to deploy the flexfield to the mainline metadata
- **Add Segment and Edit Segment** tools for descriptive flexfields, use the Save and Deploy command to save your changes. Then deploy the flexfield to the mainline metadata

Once deployed, the deployment status indicates the state of the currently configured flexfield relative to the last deployed definition.

Optionally Changing a Flexfield Segment Appearance
The flexfield attributes that you define integrate with the user interface pages where users access the attributes' business object. Application development determines the UI pages where business objects appear and the display patterns used by default to render flexfield segments.

After a flexfield has been deployed to the mainline MDS repository so that it appears on application pages, you can modify it on a per-page basis using Page Composer. For example, you can hide a segment, change its prompt or other properties, or reorder the user-defined global attributes so that they are interspersed with the core attributes in the same parent layout. You can modify the appearance of descriptive and extensible flexfield segments in the UI page using Page Composer, once the flexfield is deployed to the mainline metadata.

If the applications are running in different locales, you can provide different translations for translatable text, such as prompts and descriptions. Enter translations using the locale that requires the translated text. In the global header, click your user name and from the **Settings and Actions** menu, select **Set Preferences**. Then change the text to the translated text for that locale.

Identifying Flexfields on a Run Time Page
The **Highlight Flexfields** command in the Administration menu of the Setup and Maintenance work area identifies the location of flexfields on the run time page by displaying an **Information** icon button for accessing details about each flexfield.

Even if a descriptive or extensible flexfield isn't yet deployed and no segments appear on the run time page in normal view, the flexfield appears in the Highlight Flexfield view for that page. For descriptive flexfields, the segments as of the
last deployment appear. For extensible flexfields, any segments and contexts that have been saved but not yet deployed also appear as disabled.

**Highlight Flexfields** accesses the current flexfield metadata definition. Use the highlighted flexfield’s **Configure Flexfield** icon button to manage flexfields directly. Alternatively, note a highlighted flexfield’s name to search for it in the tasks for managing flexfields.

For more information about creating flexfields and adding them to a UI page, see the Oracle Fusion Applications Developer’s Guide. For more information about modifying flexfield segment appearance with Page Composer, see guidance on modifying existing pages in the Oracle Applications Cloud Configuring and Extending Applications guide.

**Flexfield Segment Properties**

Independent of the *value set* assigned to a *segment*, segments may have properties that affect how they’re displayed and how they function.

The following aspects are important in understanding

- Display properties
- Properties related to segment values
- Properties related to search
- Range validation segments
- Rule validation of segment values
- Naming conventions

**Display Properties**

The following table summarizes display properties.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>Whether the segment can be used.</td>
</tr>
<tr>
<td>Sequence</td>
<td>The order the segment appears in relation to the other configured segments.</td>
</tr>
<tr>
<td>Prompt</td>
<td>The string to be used for the segment’s label in the user interface.</td>
</tr>
<tr>
<td>Display type</td>
<td>The type of field in which to display the segment.</td>
</tr>
<tr>
<td>Selected and deselected values</td>
<td>If the display type is check box, the actual values to save. For example, Y and N or 0 and 1.</td>
</tr>
<tr>
<td>Display size</td>
<td>The character width of the field.</td>
</tr>
<tr>
<td>Display height</td>
<td>The height of the field as measured in visible number of lines when the display type is a text area.</td>
</tr>
<tr>
<td>Read only</td>
<td>Whether the field should display as read-only, not editable text.</td>
</tr>
</tbody>
</table>
Properties Related to Search
Extensible flexfield segments can be marked as selectively required in search using the indexed property. The indexed property requires users to enter a value before conducting a search on the attribute represented by the indexed segment. A database administrator must create an index on the segment column representing the indexed attribute.

Range Validation of Segments
Range validation enables you to enforce an arithmetic inequality between two segments of a flexfield. For example, a product must be ordered before it can be shipped. Therefore, the order date must be on or before the ship date. Also, the order date segment value must be less than or equal to the ship date segment value. You can use range validation to ensure this relationship.

The conditions for range validation are as follows:

- Segments must be configured for range validation in pairs, one with the low value and one with the high value.
- Both segments must be of the same data type.
- Both segments must be parts of the same structure in a key flexfield or parts of the same context in a descriptive flexfield or extensible flexfield.
- The low value segment must have a sequence number that's lesser than that of the high value segment.
- Non-range validated segments can exist between a range validated pair, but range validated pairs can't 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 detect a low value segment first, and the next range validated segment that it detects must be a high value segment. These two segments are assumed to be a matching pair. The low value and the high value can be equal.

Rule Validation of Segment Values
Validation rules on descriptive and extensible flexfield segments determine how an attribute is validated. The value entered for an attribute on a business object must match a specified format or be restricted to a list of values. You can use a value set or a Groovy validator to specify the validation rules.

Value set validation is required for global segments and context-sensitive segments, and optional for context segments. In the case of context segments, the application may validate a value instead of the value set validating the value against the context segment. However the application entered values must match exactly the valid context segment values. If the context segment values are a superset or subset of the input values, you must assign a table-validated value set or independent value set to validate context values.
You can also use Groovy validation to set additional restrictions or requirements for what values are allowed for certain attributes of business objects. This is useful when you need to use the same value set to validate multiple segments, but the exact validation requirement changes with each case. These validators can be defined at the global segment level, or at the context level, based on your business needs. They have a validator code, validation expression, error message, and description. After adding a new validator, click the Groovy Expression Builder icon to open the expression builder window where you define your validation expression. Groovy validation is done when a user tries to save their values to an attribute that has a Groovy validator. If the value for this attribute fails validation against the Groovy expression, the text defined in the Error Message column is displayed as an error message.

When you configure a descriptive flexfield segment, you can specify a constant to use for setting the initial value. The initial value can be an available parameter. For every planned segment, list the constant value or parameter, if any, to use for the initial value.

Naming Conventions
Enter a unique code, name, and description for the segment. These properties are for internal use and not displayed to end users. You can't change the code after the segment is created.

The Application Programming Interface (API) name is a name for the segment that isn't exposed to users. The API name is used to identify the segment in various integration points including web services, rules, and business intelligence. Use alphanumeric characters only with a leading character. For example, enter a code consisting of the characters A-Z, a-z, 0-9 with a non-numeric leading character. The use of spaces, underscores, multi-byte characters, and leading numeric characters isn't permitted. You can't change the API name after the segment has been created.

Naming Conventions for Flexfield APIs
Application Programming Interface (API) name is a name for your flexfield component, which isn't exposed to users. These names are used to identify flexfield components in various integration points, including web services, rules, and business intelligence.

How to Frame API Names
You must use only alphanumeric characters for API names. For example, enter a name with the characters A-Z, a-z, or 0-9, with a non-numeric leading character. Don't use spaces, underscores, multi-byte characters, or leading numeric characters in your API names.

⚠️ Caution: Do not change API names after they're created. Doing so could break integration points.

Words You Can't Use for API Names
You can't use certain words for API names when configuring flexfields because they're reserved words in ADF, SQL, PL/SQL, Java, Groovy, and so on.

This table lists some of the reserved words that can't be used as API names. This list isn't exhaustive.

<table>
<thead>
<tr>
<th>Letter</th>
<th>Reserved Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>ABORT, ABSTRACT, ACCEPT, ACCESS, ACTIONENABLED, ADD, ALL, ALLROWSINRANGE, ALTER, AND, ANY, ARRAY, ARRAYLEN, AS, ASC, ASSERT, ASSIGN, AT, ATTRIBUTEDEF, ATTRIBUTEDEFS, ATTRIBUTEVALUE, ATTRIBUTEVALUES, AUDIT, AUTHORIZATION, AVG</td>
</tr>
<tr>
<td>B</td>
<td>BEGIN, BETWEEN, BINDINGS, BODY, BOOLEAN, BREAK, BY, BYTE, BYVALUE</td>
</tr>
<tr>
<td>Letter</td>
<td>Reserved Words</td>
</tr>
<tr>
<td>--------</td>
<td>----------------</td>
</tr>
<tr>
<td>C</td>
<td>CASCADE, CASE, CAST, CATCH, CATEGORY, CHAR, CHECK, CHILDREN, CLASS, CLONE, CLOSE, CLUSTER, CLUSTERS, COALESCE, COLAUTH, COLUMN, COLUMNS, COMMENT, COMMIT, COMPRESS, CONNECT, CONST, CONSTANT, CONSTRIANT, CONTAINS, CONTINUE, COUNT, CRASH, CREATE, CURRENT, CURRENTROW, CURRVAL, CURSOR</td>
</tr>
<tr>
<td>D</td>
<td>DATABASE, DATACONTROL, DATE, DBA, DEBUGOFF, DEBUGON, DECIMAL, DECLARE, DEFAULT, DEFINITION, DELAY, DELETE, DELTA, DESC, DESCRIPTION, DIGITS, DISPLAY, DISPLAYDATA, DISPLAYHINT, DISPLAYHINTS, DISPOSE, DISTINCT, DIV, DO, DOUBLE, DROP</td>
</tr>
<tr>
<td>E</td>
<td>ELSE, ELSIF, EMPTY, ENABLED, ENABLEDSTRING, END, ENTRY, EQ, EQUALS, ERROR, ESTIMATEDROWCOUNT, EXCEPTION, EXCLUSIVE, EXISTS, EXIT, EXTENDS</td>
</tr>
<tr>
<td>F</td>
<td>FALSE, FETCH, FILE, FINAL, FINALIZE, FINALLY, FINDMODE, FLOAT, FOR, FORM, FROM, FULLNAME, FUNCTION, FUTURE</td>
</tr>
<tr>
<td>G</td>
<td>GE, GENERIC, GETCLASS, GOTO, GRANT, GROUP, GROUPBY, GT</td>
</tr>
<tr>
<td>H</td>
<td>HASHCODE, HAVING, HINTS</td>
</tr>
<tr>
<td>I</td>
<td>IDENTIFIED, IF, IMMEDIATE, IMPLEMENTS, IMPORT, IN, INCREMENT, INDEX, INDEXES, INDICATOR, INITIAL, INNER, INPUTVALUE, INSERT, INSTANCEOF, INT, INTEGER, INTERFACE, INTERSECT, INTO, IS, ITERATORBINDING</td>
</tr>
<tr>
<td>J</td>
<td>JAVA</td>
</tr>
<tr>
<td>K</td>
<td>KEY</td>
</tr>
<tr>
<td>L</td>
<td>LABEL, LABELS, LABELSET, LE, LEVEL, LIKE, LIMITED, LOCK, LONG, LOOP, LT</td>
</tr>
<tr>
<td>M</td>
<td>MANDATORY, MAX, MAXEXTENTS, MIN, MINUS, MLSLABEL, MOD, MODE, MODIFY</td>
</tr>
<tr>
<td>N</td>
<td>NAME, NATIVE, NATURAL, NE, NEW, NEXTVAL, NOAUDIT, NOCOMPRESS, NOT, NOTFOUND, NOTIFY, NOTIFYALL, NOWAIT, NULL, NULLIF, NUMBER</td>
</tr>
<tr>
<td>O</td>
<td>OF, OFFLINE, ON, ONLINE, OPEN, OPERATIONENABLED, OPERATOR, OPTION, OR, ORDER, ORDERBY, OTHERS, OUT, OUTER, OVERLAPS</td>
</tr>
<tr>
<td>P</td>
<td>PACKAGE, PARTITION, PCTFREE, POSITIVE, PRAGMA, PRIMARY, PRIOR, PRIVATE, PRIVILEGES, PROCEDURE, PROTECTED, PUBLIC</td>
</tr>
<tr>
<td>R</td>
<td>RAISE, RANGE, RANGEFOR, RANGESIZE, RANGESTART, RAW, REAL, RECORD, REFERENCES, RELEASE, REMR, RENAME, RESOURCE, REST, RESULT, RETURN, REVERSE, REVOKE, ROLLBACK, ROOTNODEBINDING, ROW, ROWID, ROWLABEL, ROWNUM, ROWS, ROWTYPE, RUN</td>
</tr>
</tbody>
</table>
How Flexfields Segments are Rendered
Flexfield segments appear on pages as attributes of business objects.

Settings That Affect Flexfield Segment Display
When you configure flexfield segments, the value you enter for the segment’s display type determines how the segment appears at run time.

How Display Type Values Appear
The following series of figures (A to K) represent how the display types render on the UI at run time. Each display type screenshot is assigned an alphabet that maps to the display type and its description in the table.

<table>
<thead>
<tr>
<th>Letter</th>
<th>Reserved Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>SAVEPOINT, SCHEMA, SELECT, SELECTEDVALUE, SEPARATE, SESSION, SET, SHARE, SHORT, SIZE, SMALLINT, SPACE, SQL, SQLBUF, SQLCODE, SQLERRM, START, STATEMENT, STATIC, STDDEV, SUBTYPE, SUCCESSFUL, SUM, SUPER, SWITCH, SYNCHRONIZED, SYNONYM, SYSDATE</td>
</tr>
<tr>
<td>T</td>
<td>TABAUTH, TABLE, TABLES, TASK, TERMINATE, THEN, THIS, THROW, THROWS, TO, TOOLTIP, TOSTRING, TRANSIENT, TRIGGER, TRUE, TRY, TYPE</td>
</tr>
<tr>
<td>U</td>
<td>UID, UNION, UNIQUE, UPDATE, UPDATEABLE, USE, USER</td>
</tr>
<tr>
<td>V</td>
<td>VALIDATE, VALUES, VAR, VARCHAR, VARCHAR2, VARIANCE, VIEW, VIEWS, VOID, VOLATILE</td>
</tr>
<tr>
<td>W</td>
<td>WAIT, WHEN, WHENEVER, WHERE, WHILE, WITH, WORK</td>
</tr>
<tr>
<td>X</td>
<td>XOR</td>
</tr>
</tbody>
</table>
The following figure contains the representation of a check box, a drop-down list, a list of values, and a search enabled list of values.

A. Check Box

![Check Box Image]

B. Drop-down List

![Drop-down List Image]

C. List of Values

![List of Values Image]

D. Search Enabled List of Values

![Search Enabled List of Values Image]

The following figure contains the representation of a radio button group, text area, text box, date and time, and rich text editor.
This figure contains the representation of a color palette and a static URL field.
The following table describes each display type.

<table>
<thead>
<tr>
<th>Figure Reference</th>
<th>Display Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Check Box</td>
<td>The field appears as a check box. If the user selects the check box, the checked value is used. Otherwise, the deselected value is used.</td>
</tr>
<tr>
<td>B</td>
<td>Drop-down List</td>
<td>The field appears as a list of values available to the user for selection.</td>
</tr>
<tr>
<td>C</td>
<td>List of Values</td>
<td>The field appears as a list of values available to the user for selection. The user can also click Search to find more values.</td>
</tr>
<tr>
<td>D</td>
<td>Search Enabled List of Values</td>
<td>The field appears as a text field with a Search icon button. The users can type a value in the text field or they can click</td>
</tr>
<tr>
<td>Figure Reference</td>
<td>Display Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>the Search icon button to open another window for searching.</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>Radio Button Group</td>
<td>The field appears as a set of radio buttons. The user can select one button. Selecting a button deselects any previously selected button in the set.</td>
</tr>
<tr>
<td>F</td>
<td>Text Area</td>
<td>The field appears as a text area in which the user can type multiple lines of text. The display width and height specify the visible width and number of lines in the text area, respectively.</td>
</tr>
<tr>
<td>G</td>
<td>Text Box</td>
<td>The field appears as a text field in which the user can type a single line of text. The display width controls the width of the text box.</td>
</tr>
<tr>
<td>H</td>
<td>Date Time</td>
<td>The field enables the user to enter a date if the data type is Date, or a date and time if the data type is Date Time. The user can select the date in a calendar. If the data type is Date Time, the field also displays fields for specifying the hour, minutes, seconds, AM or PM, and time zone.</td>
</tr>
<tr>
<td>I</td>
<td>Rich Text Editor</td>
<td>The field appears as a text area in which the user can enter and edit multiple lines of formatted text. The display width and height specify the visible width and number of lines in the rich text editor, respectively.[Note: This display type is available for extensible flexfields only.]</td>
</tr>
<tr>
<td>J</td>
<td>Color</td>
<td>The field displays a color palette for the user to select a color at run time and assign it to the segment. During setup, this display type appears in the list for selection only if:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• You are working on an extensible flexfield segment.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The value set for the segment is set to <code>ORA_FND_COLOR_#RRGGBB</code>.</td>
</tr>
<tr>
<td>K</td>
<td>Static URL</td>
<td>The field appears as a text field in which users can enter a fixed URL that opens the web page when clicked.</td>
</tr>
</tbody>
</table>
How Flexfields and Value Sets Work Together

Value sets are specific to your enterprise. When gathering information using flexfields, your enterprise's value sets validate the values that your users enter based on how you defined the value set.

You can assign a value set to any number of flexfield segments in the same or different flexfields. Value set usage information indicates which flexfields use the value set.

The following aspects are important in understanding how flexfields and value sets work together:

- Defining value sets
- Shared value sets
- Deployment

Defining Value Sets

As a key flexfield guideline, define value sets before configuring the flexfield, because you assign value sets to each segment as you configure a flexfield. With descriptive and extensible flexfields, you can define value sets when adding or editing a segment.

Note: Ensure that changes to a shared value set are compatible with all flexfield segments that use the value set.

Shared Value Sets

When you change a value in a shared value set, the change affects the value set for all flexfields that use that value set. The advantage of a shared value set is that a single change propagates to all usages. The drawback is that the change shared across usages may not be appropriate in every case.

Value Set Values

To configure user-defined attributes to be captured on the value set values in the Manage Value Sets task, configure the Value Set Values descriptive flexfield. The object's code is FND_VS_VALUES_B. This flexfield expects the context code to correspond to the value set code. For each value set, you can define a context whose code is the value set code, and whose context-sensitive segments are shown for the values of that value set. By default, the context segment is hidden since it maps to the value set code and is not expected to be changed.

You can also define global segments that are shown for all value sets. However, this would be quite unusual since it would mean that you want to capture that attribute for all values for all value sets.

Deployment

When you deploy a flexfield, the value sets assigned to the segments of the flexfield provide users with the valid values for the attributes represented by the segments.
Default Segment Values

To populate a flexfield segment with a default value when a row is created, specify a default type of constant, parameter, or Groovy, and a default value or expression.

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 setting a default value or deriving a default value from a parameter, only those attributes designated by development as parameters are available for selection. Different combinations of making the segments read only or editable in combination with the default or derivation value or both, have different effects.

If your segment's default type is Groovy, you can set the Groovy expression you need using the expression builder. To open the expression builder, select Groovy Expression as your Default Type and click the Groovy Expression Builder icon. But you should know that Groovy defaulting doesn't support derivation when a dependent parameter changes. The expression is evaluated only at segment creation.

The following table maps these different combinations. Initial runtime action corresponds to the row for the attribute value being created in the entity table. If the default value is read only, it can't subsequently be changed through the user interface. If the default value isn't read only, users can modify it. However, if the segment value is a derived value, a user-modified segment value is overwritten when the derivation value changes.

<table>
<thead>
<tr>
<th>Default Type</th>
<th>Default value specified?</th>
<th>Derivation value specified?</th>
<th>Initial runtime action</th>
<th>Runtime action after parameter changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>No</td>
<td>Yes</td>
<td>No initial segment value</td>
<td>The changed parameter derivation value updates segment value</td>
</tr>
<tr>
<td>Constant</td>
<td>Yes</td>
<td>No</td>
<td>Default segment value</td>
<td>N/A</td>
</tr>
<tr>
<td>Constant</td>
<td>Yes</td>
<td>Yes</td>
<td>Default segment value</td>
<td>The changed parameter derivation value updates segment value</td>
</tr>
<tr>
<td>Parameter</td>
<td>Yes</td>
<td>No</td>
<td>The default segment value is the parameter's default value</td>
<td>N/A</td>
</tr>
<tr>
<td>Parameter</td>
<td>Yes</td>
<td>Yes, and same as default value</td>
<td>The default segment value is the parameter's default and derivation value</td>
<td>The changed parameter derivation value updates segment value</td>
</tr>
<tr>
<td>Parameter</td>
<td>Yes</td>
<td>Yes, and different from default value</td>
<td>The default segment value is the parameter's default value</td>
<td>The changed parameter default value doesn't update segment value. Only the changed derivation</td>
</tr>
</tbody>
</table>
### Flexfield Usages

The flexfield usage specifies the table with which the flexfield and its segments are associated. A flexfield can have multiple usages. However, the first table registered for a flexfield indicates the master usage. Segments are based on the master usage. Other usages of the same table for the same flexfield use the same segment setup, though the column names may have a differentiating prefix.

On the Manage Descriptive Flexfields and Manage Extensible Flexfields pages, click the **Show Entity Usages** icon for a specific flexfield to view its entity usage. On the Manage Value Sets page, you can view the flexfield usages for a selected value set.

### Extensible Flexfields

For extensible flexfield contexts, you can configure a different usage. The use of an extensible flexfield context determines the scenarios or user interfaces in which the segments of a context appear to users. For example, the Supplier page displays an extensible flexfield’s supplier usage and the Buyer page for the same flexfield displays the buyer usage. Then, a context that is associated only with the supplier usage appears only on the Supplier page and not on the Buyer page.

### Value Sets

The use of value sets specifies the flexfields having segments where the identified value set is assigned.
FAQs for Flexfield Management

How can I access predefined flexfields?
Search for predefined flexfields using the manage flexfields tasks.

1. In the Setup and Maintenance work area, go to any of the following tasks:
   - Manage Descriptive Flexfields
   - Manage Extensible Flexfields
   - Manage Key Flexfields

2. On the page for the type of flexfield you're looking for, enter any of the search parameters and click **Search**.
   - **Tip:** If you don't know the flexfield name or the code, use the **Module** field to filter search results.

3. Click a flexfield to view its details.

For configuration that's not available through the manage flexfields tasks and the UI, contact My Oracle Support at https://support.oracle.com.

**Related Topics**
- Update Existing Setup Data

Why can't I edit my flexfield or value set configuration?
Your flexfield or value set configuration may be protected. Application developers mark some configurations as protected, indicating that you can't edit them.

Some examples of configurations that may be protected are:

- Descriptive flexfields
- Extensible flexfield contexts
- Extensible flexfield pages
- Value sets

Why did my page not display any flexfield?
For a flexfield to be available on the page, it must be registered by developers and also deployed. The segments appear on the page only after you have successfully deployed the flexfield.

A flexfield's deployment status indicates whether the flexfield segments are available to users. The **flexfield segments** that users see at runtime correspond to the flexfield definition last deployed successfully.

For information about registering flexfields, see the Oracle Fusion Applications Developer's Guide. Some business objects aren't designed to support flexfields. For information about how to enable business objects with flexfield capability, see Getting Started with Flexfields in the Oracle Fusion Applications Developer's Guide.

- **Note:** Oracle CX Sales and Oracle B2B Service don't support flexfields.

To add attributes to these applications, you may use Application Composer. For more information, see the product-specific documentation.
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 Oracle Metadata Services (MDS) Repository when the flexfield is deployed, are cached within a user session. You must sign out and sign back in again to view flexfield definition changes reflected in the runtime application user interface page.

How can I enable flexfield segments for Oracle Social Network Cloud Service?
When you manage Oracle Social Network Objects during setup and maintenance, search for the business object that includes descriptive flexfields. Select the attributes that are defined as flexfield segments and enable them.

Flexfield Deployment

Overview of Flexfield Deployment
Deployment generates or refreshes the Application Development Framework (ADF) business component objects that render the flexfield in a user interface. The deployment process adds user-defined attributes to the Web Services Description Language (WSDL) schemas exposed by Oracle ADF services and used by SOA composites. Flexfields are deployed for the first time during the application provisioning process. After you configure or change a flexfield, you must deploy it to make the latest definition available to users.

If a descriptive flexfield is enabled for business intelligence, the deployment process redeploy the flexfield's business intelligence artifacts.

You can deploy a flexfield to a sandbox for testing or to the mainline metadata for use in a test or production run time environment. You can deploy extensible flexfields as a background process.

After deployment, the user-defined attributes are available for incorporating into the SOA infrastructure, such as business process and business rule integration. For example, you can now write business rules that depend on the user-defined attributes. You must sign out and sign back in to Oracle Applications Cloud to see the changes you deployed at run time.

The following aspects are important in understanding flexfield deployment:

- Deployment Status
- Initial Deployment Status
- Metadata Validations
- Metadata Synchronization
- Deployment as a Background Process
- Export of Artifacts from Flexfield MDS

Deployment Status
Every flexfield has a deployment status. Check the deployment status of your flexfield after patching. The following table lists the different deployment statuses a flexfield can have.

<table>
<thead>
<tr>
<th>Deployment Status</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edited</td>
<td>The flexfield metadata definition hasn't been deployed yet. Updates of the metadata definition aren't applied in the run time environment yet.</td>
</tr>
<tr>
<td>Deployment Status</td>
<td>Meaning</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Patched</td>
<td>The flexfield metadata definition has been modified through a patch or a data migration action, but the flexfield hasn't yet been deployed. So, the updated definition isn't reflected in the runtime environment.</td>
</tr>
<tr>
<td>Deployed to Sandbox</td>
<td>The current metadata for the flexfield is deployed in ADF artifacts and available as a flexfield-enabled sandbox. The status of the sandbox is managed by the Manage Sandboxes dialog box available in the Settings and Actions menu.</td>
</tr>
<tr>
<td>Deployed</td>
<td>The current metadata for the flexfield is deployed in ADF artifacts and available to users. No changes have been made to the flexfield after being deployed to the mainline metadata.</td>
</tr>
<tr>
<td>Error</td>
<td>The deployment attempt in the mainline metadata failed.</td>
</tr>
</tbody>
</table>

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

**Initial Deployment Status of Flexfields**

The Oracle Applications Cloud implementation loads flexfield metadata into the database. This initial load sets the flexfield status to Edited. During installation, the application provisioning process deploys the flexfields of the provisioned applications, setting their status to Deployed if no errors occur.

In 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. The deployment process ends if an error occurs during the metadata validation phase. Metadata validation results don't affect the deployment status of a flexfield.

**Metadata Synchronization**

When an extensible or descriptive flexfield is deployed, the deployment process regenerates the XML schema definition (XSD). As a result, the user-defined attributes are available to web services and the SOA infrastructure.

After deploying a flexfield configuration, you must synchronize the updated XML schema definition (XSD) files in the MDS repositories for each SOA application.

**Note:** To synchronize the updated XSD files in the MDS repositories in Oracle Cloud implementations, log a service request using My Oracle Support at http://support.com/

**Deployment as a Background Process**

You can deploy extensible flexfields offline as a background process and continue working in the session without having to wait for the deployment to complete. You can queue up several extensible flexfields and deploy as a background process. The flexfields are deployed, one at a time, in the order that you deploy them to the queue. You must deploy extensible flexfields with more than 30 categories as a background process.
You can remove an extensible flexfield from the deployment queue with the Cancel Background Deployment command. When an extensible flexfield is deployed in a background process, its offline status indicates that the flexfield is in a background deployment process. A flexfield's offline status is cleared and its deployment status updated when the background deployment process has completed.

Export of Artifacts from Flexfield MDS
You can export business components from MDS for descriptive, extensible, or key flexfields, mainly for use in troubleshooting issues with flexfields. Use Download Flexfield Archive on the Manage Flexfields page to export MDS artifacts of the selected flexfield, and import them to an archive on your local computer. You can use these archived business components of flexfields for troubleshooting purposes. Alternatively, export the deployed artifacts using exportMetadata WLST.

How Flexfield Deployment Status is Calculated

Flexfield deployment status indicates how the flexfield metadata definition in the Oracle Applications Cloud database relates to the Application Development Framework (ADF) business components residing in an Oracle Metadata Services (MDS) Repository.

The following aspects are important in understanding how flexfield deployment status is calculated:

- Settings that affect flexfield deployment status
- How deployment status is calculated

Settings That Affect Flexfield Deployment Status
If you have made a change to a flexfield and expect a changed deployment status, ensure that you have saved your changes. No settings affect flexfield deployment status.

How Deployment Status Is Calculated
If the flexfield definition has been edited through the Define Flexfields activity task flows, the status is Edited. The latest flexfield metadata definition 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 application matches with the metadata definition that generated ADF business components in a sandbox MDS Repository. Whether the sandbox is active or not doesn't affect the deployment status. If the flexfield was deployed to a sandbox and hasn't been edited or redeployed to the mainline metadata since then, the status remains Deployed to Sandbox independent of whether the sandbox is active, or who is viewing the status.

If you deploy the flexfield successfully to the mainline metadata, the status is Deployed. The latest flexfield metadata definition in the 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 metadata. If either type of deployment fails and that the current flexfield definition isn't deployed, the status is Error. The deployment error message gives details about the error. The latest flexfield metadata definition in the 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 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 remains at Edited to reflect that there are still changes (outside of the patch) that aren't yet in effect.

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

**How Deploying a Flexfield-Enabled Sandbox Works with Mainline Metadata**

The *flexfield* definition in a *sandbox* corresponds to the flexfield metadata definition in the Oracle Applications Cloud 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 metadata*.

A flexfield-enabled sandbox uses the following components.

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

The following figure shows the two types of deployment available in the Manage Flexfield tasks of the Define Flexfields activity. Deploying a flexfield to a sandbox creates a sandbox MDS Repository for the sole purpose of testing flexfield behavior. The sandbox is only accessible to the administrator who activates and accesses it, not to users generally. Deploying a flexfield to the mainline metadata applies the flexfield definition to the mainline MDS Repository where it is available to end users. After deploying the flexfield to the mainline metadata, modify the page where the flexfield
segments appear. Modifications done to 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.

**Caution:** Don’t modify flexfield segment display properties using Page Composer in a flexfield-enabled sandbox as these changes will be lost when deploying the flexfield to the mainline metadata.
Mainline Metadata Services Repository Data

The Oracle Fusion Applications database stores the single source of truth about a flexfield. When the flexfield is deployed, the ADF business component objects that implement the flexfield in the run time user interface are generated in the mainline MDS Repository from this source.

Related Topics
- How You Manage Configurations in Classic Sandboxes

Considerations for Deploying a Flexfield to a Sandbox

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

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

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

When you deploy a flexfield sandbox, the process generates the name of the flexfield sandbox, and that flexfield sandbox is set as your current active sandbox. When you next sign in to the application, you can see the updated flexfield configurations. The Oracle Applications Cloud global header displays your current session sandbox.

Note: Unlike a standalone sandbox created using the Manage Sandboxes dialog box, the sandbox deployed for a flexfield contains only the single flexfield. You can manage flexfield sandboxes, such as setting an existing flexfield sandbox as active or deleting it, using the Manage Sandboxes dialog box.

When you deploy a flexfield to the mainline metadata after having deployed it to the sandbox, the sandbox-enabled flexfield is automatically deleted.

Sandbox MDS Repository Data

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

Caution: Don't modify flexfield segment display properties using Page Composer in a flexfield-enabled sandbox as these changes will be lost when deploying the flexfield to the mainline metadata.

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 dialog box in the Administrator menu of the Setup and Maintenance work area to activate and access a flexfield-enabled sandbox.

Note: Whether you use the Define Flexfields or Manage Sandboxes task flows to access a flexfield-enabled sandbox, you must sign out and sign back in before you can see the changes you deployed in the run time.

You cannot publish the flexfield from the sandbox to the mainline metadata. You must use the Define Flexfields task flow pages to deploy the flexfield for access by users of the mainline metadata because the flexfield configuration in the mainline metadata is the single source of truth.
Related Topics
- How You Manage Configurations in Classic Sandboxes

Manage Value Sets

Overview of Value Sets
A value set is a group of valid values that you assign to a flexfield segment to control the values that are stored for business object attributes.

A user enters a value for an attribute of a business object while using the application. The flexfield validates the value against the set of valid values that you configured as a value set and assigned to the segment.

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

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

Note: Ensure that changes to a shared value set are compatible with all flexfields segments using the value set.

The following aspects are important in understanding value sets:
- Managing value sets
- Validation
- Security
- Precision and scale
- Usage and deployment
- Protected value set data

Managing Value Sets
To open the Manage Value Sets page, use the Manage Value Sets task. You can also use the Manage Descriptive Flexfields and Manage Extensible Flexfields tasks for configuring a segment, including its value set. To open the Manage Values page, select the value set from the Manage Value Sets page, and click Manage Values. Alternatively, click Manage Values from the Edit Value Set page.

Validation
The following types of validation are available for value sets:
- Format only, where users enter data instead of selecting values from a list
- Independent, a list of values consisting of valid values you specify
- Dependent, a list of values where a valid value derives from the independent value of another segment
- Subset, where the list of values is a subset of the values in an existing independent value set
- Table, where the values derive from a column in an application table and the list of values is limited by a WHERE clause

A segment that uses a format only value set doesn’t present a list of valid values to users. If required, you may add table validated value sets to the list of available value sets available for configuration.
Note: For the Accounting Key Flexfield value sets, you must use independent validation only. If you use other validations, you can’t use the full chart of accounts functionality, such as data security, reporting, and account hierarchy integration.

Security
Value set security only works in conjunction with usage within flexfield segments. You can specify that data security be applied to the values in flexfield segments that use a value set. Based on the roles provisioned to users, data security policies determine which values of the flexfield segment users can view or modify.

The application of value set security has the following conditions:

- At the value set level: The value set is the resource secured by data security policies. If a value set is secured, every usage of it in any flexfield is secured. Disabling security for individual usages of the same value set isn’t possible.
- Applies to independent, dependent, or table-valuated value sets.
- Applies mainly when data is being created or updated, and to key flexfield combinations tables for query purposes. Value set security doesn’t determine which descriptive flexfield data is shown upon querying.
- Security conditions defined on value sets always use table aliases. When filters are used, table aliases are always used by default. When predicates are defined for data security conditions, make sure that the predicates also use table aliases.

For key flexfields, the attributes in the view object corresponding to the account combination ID, structure instance number (SIN), and data set number (DSN) can’t be transient. They must exist in the database table. For key flexfields, the SIN segment is the discriminator attribute, and the account combination segment is the common attribute.

Precision and Scale
If the data type of a value set is Number, you can specify the precision (maximum number of digits user can enter) or scale (maximum number of digits following the decimal point).

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

Protected Value Set Data
Application developers may mark some value sets as protected, indicating that you can’t edit them.

You can edit only value sets that are not marked as protected. You can’t edit or delete protected value sets. If the value set type supports values (such as independent, dependent or subset value sets), then you can’t add, edit, or delete values.
Note: References to protected value sets aren't restricted. Value sets, protected or not, may be assigned to any flexfield segment. Likewise, other value sets may reference protected value sets; for example, an unprotected dependent value set may reference a protected independent value set.

Related Topics

- Chart of Accounts Components

Validation Type Options for Value Sets

Validation and usage of value sets determine where and how users access valid values for attributes represented by flexfield segments.

Tip: As a flexfield guideline, define value sets before configuring the flexfield, because you can assign value sets to each segment as you configure a flexfield. With descriptive and extensible flexfield segments, you can create value sets when adding or editing a segment on the runtime page where the flexfield appears.

The following aspects are important in defining value sets:

- Value sets for context segments
- Format-only validation
- Interdependent value sets
- Table validation
- Range
- Security
- Testing and maintenance

Value Sets for Context Segments

When assigning a value set to a context segment, you can only use table-validated or independent value sets.

You can use only table and independent value sets to validate context values. The data type must be character and the maximum length of the values being stored must not be larger than the context’s column length. If you use a table value set, the value set can't reference flexfield segments in the value set’s WHERE clause, other than the flexfield segment to which the value set is assigned.

Format Only Validation

The format only validation type enables users to enter any value, as long as it meets your specified formatting rules. 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 permits only numeric characters, users can enter the value 456 (for a value set with maximum length of three or more), but can't enter the value ABC. A format only value set doesn't otherwise restrict the range of different values that users can enter. For numeric values, you can also specify if a numeric value should be zero filled or how many digits should follow the radix separator.

Interdependent Value Sets

Use an independent value set to validate data against a list that isn't stored in an application table, and not dependent on a subset of another independent value set. You can't specify a dependent value set for a given segment without having first defined an independent value set that you apply to another segment in the same flexfield. Use a dependent value set to limit the list of values for a given segment based on the value that the user has defined 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 the states in the USA with values such as CA, NY, and so on. Then you define a dependent value set of cities in the USA with values such as San Francisco and Los Angeles that are valid for the independent value CA. Similarly, New York City and Albany 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. Users don’t have to select a value for another segment first to have access to the subset value set.

Independent, dependent, and subset value sets require a user-defined list of valid values. Use the Manage Values page to create and manage a value set’s valid values and the order in which they appear.

**Tip:** You can configure the Manage Value Sets page to capture additional attributes for each valid value by adding context-sensitive segments in a new context for FND_VS_VALUES_B descriptive field.

### Table Validation

Typically, you use a table-validated set when the values you want to use are already maintained in an application table, such as a table of supplier names. Specify the table column that contains the valid value. You can optionally specify the description and ID columns, a WHERE clause to limit the values to use for your set, and an ORDER BY clause.

If you specify an ID column, then the flexfield saves the ID value, instead of the value from the value column, in the associated flexfield segment. If the underlying table supports translations, you can enable the display of translated text by basing the value set’s value column on a translated attribute of the underlying table. You should also define an ID column that’s based on an attribute that isn’t language-dependent so that the value’s invariant ID (an ID that doesn’t change) is saved in the transaction table. The run time displays the corresponding translated text from the value column for the run time session’s locale.

Table validation lets you enable a segment to depend on multiple prior segments in the same context structure. You can’t reference other flexfield segments in the table-validated value set’s WHERE clause. Which means, the WHERE clause can’t reference SEGMENT.segment_code or VALUESET.value_set_code.

Table-validated value sets have unique values across the table, irrespective of bind variables. The WHERE clause fragment of the value set is considered if it doesn’t have bind variables. If it has bind variables, the assumption is that the values are unique in the value set. If you use table validated value sets for key flexfields, then you can’t use all integration options supported for key flexfields, such as:

- Data security
- Oracle Transactional Business Intelligence (OTBI)
- Extended Spread Sheet Database (ESSbase)
- Tree or hierarchy integration

To use these integration options for key flexfields, you must use independent value sets only.

### Range

In the case of format, independent, or dependent value sets, you can specify a range to limit which values are valid. You can specify a range of values that are valid within a value set. You can also specify a range validated pair of segments where one segment represents the low end of the range and another segment represents the high end of the range.

For example, you might specify a range for a format-only value set with format type Number where the user can enter only values between 0 and 100.

### Security

In the case of independent and dependent values, you can specify that data security be applied to the values in segments that use a value set. Based on the roles provisioned to users, data security policies determine which values of the flexfield segment users can view or modify.
To enable security on a value set, specify a database resource, typically the code value for the value set. Using the Manage Database Security Policies task, specify conditions, such as filters or SQL predicates, and policies that associate roles with conditions. You can use a filter for simple conditions. For more complex conditions, use a SQL predicate.

Value set data security policies and conditions differ from data security conditions and policies for business objects in the following ways:

- You can grant only read access to users. You can't specify any other action.
- When defining a condition that's based on a SQL predicate, use VALUE, VALUE_NUMBER, VALUE_DATE, VALUE_TIMESTAMP, or VALUE_ID to reference the value from a dependent, independent, or subset value set. For table value sets, use a table alias to define the table, such as &TABLE_ALIAS category=70.

When you enable security on table-validated value sets, the security rule that's 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 must not assume anything about the bind variables. Instead the whole list of values must be available and you write the rule, for example, to permit x, or to permit y and z. By default in data security, all values are denied and show only rows to which access has been provided.

Testing and Maintenance
You don't have to define or maintain values for a table-validated value set, as the values are managed as part of the referenced table or independent value set, respectively.

You can't manage value sets in a sandbox.

When you change an existing value set, the deployment status for all affected flexfields changes to Edited. You must redeploy all flexfields that use that value set to make the flexfields reflect the changes. In the UI pages for managing value sets, the value set's usages show which flexfields are affected by the value set changes.

If your application has more than one language installed, or there is any possibility that you might install one or more additional languages for your application in the future, select Translatable. This doesn't require you to provide translated values now, but you can't change this option if you decide to provide them later.

Considerations for Planning Value Sets
The value sets you create and configure depend on the valid values on the business object attributes that will use the value set. When creating value sets, you first give the value set a name and description, and then define the valid values of the set.

The following aspects are important in planning value sets:

- List of values
- Plain text
- Value ranges
- Value format specification
- Security

List of Values
You can use one of the following types of lists to specify the valid values for a segment:

- Table column
- User-defined list. Also include a sub list.
- Dependent user-defined list
If the valid values exist in a table column, use a table value set to specify the list of values. To limit the valid values to a subset of the values in the table, use a SQL WHERE clause. Table value sets also provide some advanced features, such as enabling validation depending on other segments in the same structure.

Use an independent value set to specify a user-defined set of valid values. For example, you can use an independent value set of Mon, Tue, Wed, and so forth to validate the day of the week. You can also specify a subset of an existing independent value set as the valid values for a segment. For example, if you have an independent value set for the days of the week, then a weekend subset can comprise entries for Saturday and Sunday.

Use a dependent value set when the available values in the list and the meaning of a given value depend on which independent value was selected for a previously selected segment value. For example, the valid holidays depend on which country you are in. A dependent value set is a collection of value subsets, with one subset for each value in a corresponding independent value set.

For lists of values type value sets, you can additionally limit the valid values that an end user can select or enter by specifying format, minimum value, and maximum value. For list of values type value sets, you can optionally implement value set data security. If the applications are running in different locales, you might need to provide different translations for the values and descriptions.

Plain Text
Use a format-only value set when you want to allow users to enter any value, as long as that value conforms to formatting rules. For example, if you specify a maximum length of 3 and numeric-only, then end users can enter 456, but not 4567 or 45A. You can also specify the minimum and maximum values, whether to align the text to either side, and whether to zero-fill. With a format-only value set, no other types of validation are applied.

Value Ranges
You can use either a format-only, independent, or dependent value set to specify a range of values. For example, you might create a format-only value set with Number as the format type where the end user can enter only the values between 0 and 100. Or, you might create a format-only value set with Date as the format type where the end user can enter only dates for a specific year, such as a range of 01-JAN-93 to 31-DEC-93. Because the minimum and maximum values enforce these limits, you need not define a value set that contains each of these individual numbers or dates.

Value Format
Flexfield segments commonly require some kind of format specification, regardless of validation type. Before creating a value set, consider how you will specify the required format.

The following table shows options for validation type and value data type.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value data type</td>
<td>Character, Number, Date, Date Time.</td>
</tr>
<tr>
<td>Value subtype</td>
<td>Text, Translated text, Numeric digits only, Time (20:08), Time (20:08:08).</td>
</tr>
<tr>
<td></td>
<td>An additional data type specification for the Character data type for the Dependent, Independent, and Format validation types.</td>
</tr>
<tr>
<td>Maximum length</td>
<td>Maximum number of characters or digits for Character data type.</td>
</tr>
<tr>
<td>Precision</td>
<td>Maximum number of digits the user can enter.</td>
</tr>
</tbody>
</table>
## Define Applications Core Configuration

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale</td>
<td>Maximum number of digits that can follow the decimal point.</td>
</tr>
<tr>
<td>Uppercase only</td>
<td>Lowercase characters automatically changed to uppercase.</td>
</tr>
<tr>
<td>Zero fill</td>
<td>Automatic text alignment and zero-filling of entered numbers (affects values that include only the digits 0-9).</td>
</tr>
</tbody>
</table>

**Note:** You cannot change the text value data type to a translated text value subtype after creating a value set. If there is any chance you may need to translate displayed values into other languages, choose Translated text. Selecting the Translated text subtype doesn't require you to provide translated values.

### Value Sets for Context Segments

You can use only table and independent value sets to validate context values. The data type must be character and the maximum length of the values being stored must not be larger than the context's column length. If you use a table value set, the value set cannot reference flexfield segments in the value set's WHERE clause other than the flexfield segment to which the value set is assigned.

### Security

When enabling security on a value set, the data security resource name is an existing value set or one that you want to create. The name typically matches the code value for the value set. You cannot edit the data security resource name after you save your changes.

### Considerations for Bind Variables in Table-Validated Value Sets

After you assign a value set to a flexfield, you can use bind variables in the WHERE clause.

These bind variables refer to flexfield elements:

- `:SEGMENT.<segment_code>`
- `:CONTEXT.<context_code>;SEGMENT.<segment_code>`
- `:VALUESET.<value_set_code>`
- `:FLEXFIELD.<internal_code>`
- `:PARAMETER.<parameter_code>`

### Segment Code

`:SEGMENT.<segment_code>`

This bind variable refers to the ID or value of a segment where `<segment_code>` identifies the segment. Where referring to the ID, the value set is ID-validated. Where referring to the value, the value set isn't ID-validated. The data type of the bind value is the same as the data type of the segment’s column.

For both descriptive and extensible flexfields, the segment must be in the same context as the source segment. The source segment contains the WHERE clause. For descriptive flexfields, if the segment is global, then the source segment must be global.
The segment must have a sequence number that's less than the sequence number of the target segment with this bind variable. A matching segment must exist in the current flexfield context.

This bind variable is useful when the set of valid values depends on the value in another segment. For example, the values to select from a CITIES table might depend upon the selected country. If SEGMENT1 contains the country value, then the WHERE clause for the CITIES table might be `<country_code> = :{SEGMENT.SEGMENT1}`.

**Context Code**

`:\{CONTEXT.<context_code>;SEGMENT.<segment_code}\}`

This bind variable, which is valid only for extensible flexfields, refers to the ID (if the value set is ID-validated) or value (if not ID-validated) of a segment that's in a different context than the target segment (the segment with the WHERE clause).

- The `<context_code>` identifies the context and must be in the same category or in an ancestor category. It can't be a multiple-row context.
- The `<segment_code>` identifies the segment. The data type of the bind value is the same as the data type of the segment's column.

**Note:** The target segment should appear in the UI after the source segment to ensure the source segment has a value. If the target segment's context is a single-row context, the source and target segments must be on separate pages and the target page must follow the source page.

The framework of extensible flexfields doesn't perform any additional validation related to mismatched values for segments defined with cross context bind parameters. Administrators must populate the correct pair of segment values.

This bind variable is useful when the set of valid values depends on the value of a segment in another context. For example, the values to select from a CERTIFICATION table for a segment in the Compliance and Certification context might depend on the value of the country segment in the Manufacturing context.

**Value Set Code**

`:\{VALUESET.<value_set_code}\}`

This bind variable refers to the ID (if the value set is ID-validated) or value (if not ID-validated) of the segment that's assigned to the value set that's identified by the `<value_set_code>`. The data type of the bind value is the same as the data type of the segment's column.

The segment must have a sequence number that's less than the sequence number of the segment with this bind variable. If more than one segment is assigned to the value set, the closest prior matching segment will be used to resolve the bind expression. A matching segment must exist in the current flexfield context.

This bind variable is useful when the set of valid values depends on the value in another segment and that segment code can vary, such as when the value set is used for more than one context or flexfield. For example, the values to select from a CITIES table might depend upon the selected country. If the value set for the segment that contains the country value is COUNTRIES, then the WHERE clause for the CITIES table might be `<country_code> = :\{VALUESET.COUNTRIES\}`.

**Flexfield Internal Code**

`:\{FLEXFIELD.<internal_code}\}`
This bind variable refers to an internal code of the flexfield in which the value set is used, or to a validation date. The internal_code must be one of the following:

- **APPLICATION_ID** - the application ID of the flexfield in which this value set is used. The data type of APPLICATION_ID and its resulting bind value is NUMBER.
- **DESCRIPTIVE_FLEXFIELD_CODE** - the identifying code of the flexfield in which this value set is used. The data type of DESCRIPTIVE_FLEXFIELD_CODE and its resulting bind value is VARCHAR2. Note that you use this string for both descriptive and extensible flexfields.
- **CONTEXT_CODE** - the context code of the flexfield context in which this value set is used. The data type of CONTEXT_CODE and its resulting bind value is VARCHAR2.
- **SEGMENT_CODE** - the identifying code of the flexfield segment in which this value set is used. The data type of SEGMENT_CODE and its resulting bind value is VARCHAR2.
- **VALIDATION_DATE** - the current database date. The data type of VALIDATION_DATE and its resulting bind value is DATE.

### Flexfield Parameters

```
: {PARAMETER.<parameter_code>}
```

This bind variable refers to the value of a flexfield parameter where parameter_code identifies the parameter. The data type of the resulting bind value is the same as the parameter's data type.

**Note:** You can't assign a table value set to a context segment if the WHERE clause uses VALUESET.value_set_code or SEGMENT.segment_code bind variables.

### Create Table-Validated Value Sets Based on Lookups

In an application user interface, you want to display a list of values that customers use to enter satisfaction scores. The value column name is 1, 2, 3, 4, 5 and the value column description is Extremely Satisfied, Satisfied, and so on. Users can select the appropriate value or description which stores the corresponding name so the name value can be used in a calculation expression.

In this case, you can use the FND_LOOKUPS table as the basis for a table-validated value set. The lookup meaning corresponds to the Value Column Name and the lookup description corresponds to the Description Column Name. The following table lists the properties of the value set.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FROM clause</td>
<td>FND_LOOKUPS</td>
</tr>
<tr>
<td>WHERE clause</td>
<td><code>lookup_type = 'CN_XX_CUST_SATISFACT_SCORE'</code></td>
</tr>
<tr>
<td>ID column</td>
<td><code>lookup_code</code></td>
</tr>
<tr>
<td>Value column</td>
<td><code>meaning</code></td>
</tr>
<tr>
<td>Description column</td>
<td><code>description</code></td>
</tr>
<tr>
<td>Enable column</td>
<td><code>enabled_flag</code></td>
</tr>
</tbody>
</table>

---
Creating a Value Set Based on a Lookup

1. From the Setup and Maintenance work area, open the Manage Value Sets task and click the Go to Task icon button.
2. On the Manage Value Sets page, click the Create icon button.
3. On the Create Value Set page, enter the following values:
   a. In the Value Set Code field, enter CN_XX_CUSTOMER_SATISFACTION_SCORES
   b. In the Description field, enter Customer satisfaction score.
   c. In the Module field, select Search
   d. In the Search and Select: Module subwindow, enter Incent in the User Module Name field
   e. Select Incentive Compensation.
   f. Click OK.
4. On the Create Value Set page, enter the following values:
   a. In the Validation Type field, select Table.
   b. In the Value Data Type field, select Character.
   c. In the Definition section FROM Clause field, enter FND LOOKUPS.
   d. In the Value Column Name field, enter DESCRIPTION.
   e. In the Description Column Name field, enter MEANING.
   f. In the ID Column Name field, enter LOOKUP_CODE.
   g. In the Enabled Column Name field, enter 'Y'.
   h. In the Start Date Column Name field, enter START_DATE_ACTIVE.
   i. In the End Date Column Name field, enter END_DATE_ACTIVE.
   j. In the WHERE Clause field, enter LOOKUP_TYPE = 'CN_XX_CUST_SATISFACT_SCORE'.
5. Click Save and Close.
6. In the Manage Value Sets page, click Done.

Add Attributes to the Manage Value Sets Page

You can add attributes to independent, dependent, and subset value sets. The attributes appear on the Manage Value Sets page where you can store additional information about each valid value. To display attributes on an application page, you must programmatically modify the application.

To add attributes and subsequently view them on the Manage Value Sets page, perform the following steps:

1. Using the Manage Descriptive Flexfields task, find the FND_VS_VALUES_B flexfield and open it for editing.
2. Click Manage Contexts.
3. Create a new context and use the value set code for the context code.
4. Add new attributes as context-sensitive segments and save the changes.
5. Deploy FND_VS_VALUES_B to run time.
6. Sign out and sign back in.
7. Open the Manage Value Sets page to view the new attributes.

Import Value Set Values
You can import a file containing values that you want to edit or add to a given independent or dependent value set. For example, uploading a hundred values may be more efficient than creating them individually using the Manage Value Sets task. However, for just a few values, it may be quicker to perform the relevant tasks.

Importing Value Set Values
To import value set values:

1. Create a flat file containing the values in the value set that you want to add or update.

   **Note:**
   - When creating the file, you must specify an existing value set code to which you want to add values or edit existing values. If the value set does not exist, add the value set using the appropriate Manage Value Sets setup task in the Setup and Maintenance work area.
   - The file that you create must adhere to the formatting and content requirements for creating flat files containing value set values.

2. Upload the flat file to the content repository using the File Import and Export page.
3. Import the file using the appropriate Manage Value Sets setup task in the Setup and Maintenance work area. To import the file:
   a. Click **Actions > Import** in the Manage Value Sets page.
   b. In the File Name field, enter the name of the flat file you uploaded using the File Import and Export page.
   c. In the Account field, select the user account containing the flat file.
   d. Click **Upload**.
      **Note:** Alternatively, you can import the file using either of the following methods:
      - Run the Upload Value Set Values **scheduled process**.
      - Use the Applications Core Metadata Import web service. For more information on the Applications Core Metadata Import web service, see the SOAP Web Services guide for your cloud services.

Related Topics
- **Overview of Files for Import and Export**

Import Value Set Values Using Web Services
As an alternative to using the import option on the Manage Value Sets page, you can use the FndManageImportExportFilesService web service to import value set values into an application. The web service uses the valueSetValuesDataLoader method.
Prerequisite

Ensure that the web service FndManageImportExportFilesService is added to the Manage Application Flexfield Value Set entitlement. Use the Security Console to perform this configuration on Oracle Entitlements Server. While configuring the web service, provide the following details:

- Resource type: WebserviceResourceType
- Display name: ImportExport
- Name: http://xmlns.oracle.com/oracle/apps/fnd/applcore/webservices/FndManageImportExportFilesService#
- Entitlement (In Default Policy Domain): Manage Application Flexfield Value Set
- Security policy: oracle/wss11_saml_or_username_token_with_message_protection_service_policy

Importing Value Set Values

To import value set values:

1. Perform the following substeps to generate keystore for the security policy "oracle/wss11_saml_or_username_token_with_message_protection_service_policy":
   b. Access the WSDL using the URL http://<host>:<port>/fndAppCoreServices/FndManageImportExportFilesService?wsdl through a web browser, and get the public key <wsdl:service>/<wsdl:port>/<wsa:EndpointReference>/<wsid:Identity>/<dsig:keyInfo>/<dsig:X509Data>/<dsig:X509Certificate>. Then, enclose it with ---- BEGIN CERTIFICATE ---- and ---- END CERTIFICATE ---- and save it to a file by name cdrmpk.cer.
   c. Store the key information in the truststore using the command keytool -importcert -alias cdrmkey -file cdrmpk.cer -keystore mycompclient-keystore.jks -storepass <password>.
2. Access the WSDL page using the browser of your choice.
3. Export and save the associated security certificates Verisign Class 3 Public Primary Certification Authority - G5 and Verisign Secure Server CA - G3.
4. Use the following command to import the saved certificates into the trust store of the client computer.
   keytool -importcert -keystore <truststore> -storepass <truststorepassword> -file <file location where the mentioned certificate is stored> -alias <alias for certificate>
5. Run the following command to generate the JAX-WS proxy for the FndManageImportExportFilesService web service.
   C:\Program Files\Java\jdk1.7.0_04\bin>wsimport -s "d:\wsimport\FndManageImportExport" -d "d:\wsimport\FndManageImportExportFilesService?wsdl" FndManageImportExportFilesService?wsdl
   parsing WSDL...
   Generating code...
   Compiling code...
6. Save the generated code as a JAR file and name it FndManageImportExportProxy.jar.
7. Use the following code to create another JAR file to initiate the web service:
   package com.oracle.xmlns.oracle.apps.fnd.applcore.webservices;
   import com.sun.xml.ws.developer.WSBindingProvider;
import java.io.File;
import java.io.IOException;
import java.util.List;
import java.util.Map;
import javax.xml.ws.BindingProvider;
import javax.xml.ws.WebServiceRef;
import javax.xml.ws.handler.Handler;
import oracle.webservices.ClientConstants;
import weblogic.wsee.jws.jaxws.owsm.SecurityPoliciesFeature;

// !THE CHANGES MADE TO THIS FILE WILL BE DESTROYED IF REGENERATED!
// This source file is generated by Oracle tools
// Contents may be subject to change
// For reporting problems, use the following
// Version = Oracle WebServices (11.1.1.0.0, build 130224.1947.04102)

public class FndManageImportExportFilesServiceSoapHttpPortClient
{
    @WebServiceRef
    private static FndManageImportExportFilesService_Service fndManageImportExportFilesService_Service;

    public static void main(String[] args)
    {
        System.setProperty("javax.net.ssl.trustStore","<location of truststore which is used in II to import
        the certificates>");
        System.setProperty("javax.net.ssl.trustStorePassword", "<truststore password>");

        fndManageImportExportFilesService_Service = new FndManageImportExportFilesService_Service();
        SecurityPoliciesFeature securityFeatures = new SecurityPoliciesFeature(new String[]{ "oracle/
        wss11_username_token_with_message_protection_client_policy"},
        );
        FndManageImportExportFilesService
        fndManageImportExportFilesService = fndManageImportExportFilesService_Service.getFndManageImportExportFilesServiceSoapHttpPort
        (securityFeatures);
        // Add your code to call the desired methods.

        WSBindingProvider wsbp = (WSBindingProvider) fndManageImportExportFilesService;
        Map<String, Object> requestContext = wsbp.getRequestContext();

        requestContext.put(ClientConstants.WSSEC_KEYSTORE_TYPE,"jks");
        // Provide location of 'mycompclient-keystore.jks' which was created during Step I
        requestContext.put(ClientConstants.WSSEC_KEYSTORE_LOCATION,"/home/user1/mycompclient-keystore.jks");
        requestContext.put(ClientConstants.WSSEC_KEYSTORE_PASSWORD,"<password>");
        requestContext.put(ClientConstants.WSSEC_RECIPIENT_KEY_ALIAS,"cdrmkey");

        //Provide user who is having permission to initiate the service
        requestContext.put(WSBindingProvider.USERNAME_PROPERTY,"<username>");
        requestContext.put(WSBindingProvider.PASSWORD_PROPERTY,"<password>");

        String id = invokeUploadFiletoUCMMethod(fndManageImportExportFilesService);

        if (id != null) {
            invokevalueSetValuesDataLoader(fndManageImportExportFilesService, new Long(id));
        }
    }
}
static String invokeUploadFiletoUCMMethod(FndManageImportExportFilesService fndManageImportExportFilesService) {

    String response = null;
    DocumentDetails document = new DocumentDetails();
    ObjectFactory objfactory = new ObjectFactory();
    document.setFileName(objfactory.createDocumentDetailsFileName("import_data.txt"));
    // Provide UCM repository - if repository is fin/tax/import then suffix each value with $ as mentioned here
    document.setDocumentTitle(objfactory.createDocumentDetailsDocumentTitle("VS"));
    document.setContentType(objfactory.createDocumentDetailsContentType("plain/text"));

    try {
        // Provide location of 'VS.txt' which contains ValueSet values data in prescribed format
        byte[] content = org.apache.commons.io.FileUtils.readFileToByteArray(new File("/home/user1/VS.txt"));

        //System.out.println("File content:" + new String(content, "UTF-8"));
        document.setContent(objfactory.createDocumentDetailsContent(content));
    } catch (IOException e) {
        System.out.println(e.getMessage());
    }

    catch(Exception e) {
        System.out.println("Exception: "+e.getMessage());
    }

    try {
        response = fndManageImportExportFilesService.uploadFiletoUCM(document);
        System.out.println("Response: " + response);
    } catch (ServiceException e) {
        System.out.println(e.getMessage());
    }

    return response;
}

static void invokeValueSetValuesDataLoader(FndManageImportExportFilesService fndManageImportExportFilesService, Long id) {
    String response;
    try {
        response = fndManageImportExportFilesService.valueSetValuesDataLoader(id);
        System.out.println("Response: " + response);
    } catch (ServiceException e) {
        System.out.println(e.getMessage());
    }
}

8. Save the generated output as a JAVA file and name it FndManageImportExportFilesServiceSoapHttpPortClient.java.
9. Use the JAVA file to build a JAR file, and name it FndManageImportExportClient.jar.
10. Use the following command to run the web service:

    java -cp ./FndManageImportExportProxy.jar:./ws.api_1.1.0.0.jar:./FndManageImportExportClient.jar
        FndManageImportExportFilesServiceSoapHttpPortClient
Requirements for Flat Files to Upload Value Set Values

You can import large volumes of value set value data from the content repository. To upload value set values to the content repository, create a flat file containing the values in the value set that you want to add or update. You upload these flat files to the content repository using the File Import and Export page.

General Requirements

The first line of the flat file must contain the column names for the value set value data, including all mandatory columns, and separated by the '|' (pipe) character. Each subsequent line should contain a row of data specified in the same order as the column names, also separated by the '|' character.

The requirements for creating flat files vary with the type of value sets:

- Independent value sets
- Dependent value sets

Independent Value Set

A flat file for uploading values for independent value sets must contain the mandatory columns. The following table lists the three mandatory columns and their data types.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueSetCode</td>
<td>VARCHAR2(60)</td>
</tr>
<tr>
<td>Value</td>
<td>VARCHAR2(150)</td>
</tr>
<tr>
<td>Enabled Flag</td>
<td>VARCHAR2(1), Y or N</td>
</tr>
</tbody>
</table>

**Note:** You can also specify optional columns.

Examples:

- To upload values to a COLORS independent value set with the minimum columns, you can use the following flat file:
  
  ```
  ValueSetCode | Value | EnabledFlag
  ------------|-------|-------------
  COLORS | Red | Y
  COLORS | Orange | Y
  COLORS | Yellow | Y
  ```

- To upload values to a STATES independent value set with more (optional) columns, you can use the following flat file:
  
  ```
  ValueSetCode | Value | Description | EnabledFlag
  ------------|-------|-------------|-------------
  STATES | AK | Alaska | Y
  STATES | CA | California | Y
  STATES | WA | Washington | Y
  ```
Dependent Value Sets
A flat file for uploading values for dependent value sets must contain the mandatory columns. The following table lists the four mandatory columns and their data types.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value Set Code</td>
<td>VARCHAR2(60)</td>
</tr>
<tr>
<td>Independent Value</td>
<td>VARCHAR2(150)</td>
</tr>
<tr>
<td>Value</td>
<td>VARCHAR2(150)</td>
</tr>
<tr>
<td>Enabled Flag</td>
<td>VARCHAR2(1), Y or N</td>
</tr>
</tbody>
</table>

**Note:** You can also specify optional columns.

Example:
To upload values to a CITIES dependent value set (dependent on the STATES independent value set), you can use the following flat file:

```
ValueSetCode | IndependentValue | Value     | EnabledFlag
-------------|------------------|-----------|
CITIES       | AK               | Juneau    | Y
CITIES       | AK               | Anchorage | Y
CITIES       | CA               | San Francisco | Y
CITIES       | CA               | Sacramento | Y
CITIES       | CA               | Los Angeles | Y
CITIES       | CA               | Oakland   | Y
```

Additional Optional Columns
In addition to the mandatory columns, you can add optional columns. The following table lists the optional columns for both dependent and independent value sets.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translated Value</td>
<td>VARCHAR2(150), for use in value sets that are translatable</td>
</tr>
<tr>
<td>Description</td>
<td>VARCHAR2(240)</td>
</tr>
<tr>
<td>Start Date Active</td>
<td>DATE, formatted as YYYY-MM-DD</td>
</tr>
<tr>
<td>End Date Active</td>
<td>DATE, formatted as YYYY-MM-DD</td>
</tr>
<tr>
<td>Sort Order</td>
<td>NUMBER(18)</td>
</tr>
</tbody>
</table>
### Related Topics
- Overview of Files for Import and Export

### Upload Value Set Values Process
This process uploads a flat file containing **value set** values for **flexfields**. You can use the scheduled process to upload a file containing values you want to edit or add to an existing independent or dependent value set. This process is useful for adding or updating large volumes of value set value data in an automated or recurring fashion. For example, you can upload a hundred values on a recurring basis when scheduled as a recurring process. This method could be more efficient than using the Import action in the Manage Value Sets tasks in the Setup and Maintenance work area. However, for a task of uploading a hundred values, it may be quicker to use the Import action in the relevant tasks.

Run this process from the Scheduled Processes Overview page. You can run it on a recurring basis whenever the flat file in the content repository account is updated.

You must create the flat file containing the values data, and upload the flat file to the content repository using the File Import and Export page.

**Parameters**

**Flat File Name**

Enter the name of the flat file you uploaded using the File Import and Export page.

**Account**

Select the user account containing the flat file in the content repository to upload.

**Related Topics**
- Overview of Files for Import and Export
- Overview of Scheduled Processes

### Translation of Flexfield and Value Set Configurations
When you first configure a flexfield or **segment**, the translatable text that you enter, such as prompts and descriptions, is stored as the text for all installed locales. You may then provide a translation for a particular locale. If you don't provide a translation for a given locale, then the value that was first entered is used for that locale.

To translate the text for a particular locale, sign in with that locale, or in the global header, select **Settings and Actions > Personalization > Set Preferences** and specify the locale. Then, update the translatable text in the flexfield using
the Manage Descriptive Flexfields task, Manage Key Flexfields task, or Manage Extensible Flexfields task. Your modifications change the translated values only for the current session's locale.

After you complete the translations, deploy the flexfield.

You can define translations for a dependent value set or an independent value set, if the value set is of type Character with a subtype Translated text. You define the translations by setting the current session to the locale for which you want to define the translation. Then use the Manage Value Sets task to enter the translated values and descriptions for that locale.

You can define translated values for a table value set for which multiple languages are supported and that the value set's value column is based on a translated attribute of the underlying table. For more information about using multilanguage support features, see the Oracle Fusion Applications Developer's Guide.

FAQs for Manage Value Sets

What happens if a value set is security enabled?

Value set security is a feature that enables you to secure access to value set values based on the role of the user in the application.

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 application.

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 Applications Cloud data security.

How can I set a default value for a flexfield segment?

When you define or edit a flexfield segment, you pick a value from the assigned value set and set it as default.

You can set the default value for a descriptive flexfield segment to be a parameter. The mapped entity object attribute provides the initial default value for the segment.

You can set the default value to be a constant, if appropriate to the data type of the value set assigned to the segment.

In addition to an initial default value, you can set a derivation value for updating the attribute's value every time the parameter value changes. The parameter you select identifies the entity object source attribute. Any changes in the value of the source attribute during run time are reflected in the value of the segment.

If the display type of the segment is a check box, you can set whether the default value of the segment is checked or unchecked.
Manage Descriptive Flexfields

Overview of Descriptive Flexfields

Use **descriptive flexfields** to add attributes to business object entities, and define validation for them.

All the business object entities that you can use in the application are enabled for descriptive flexfields. However, configuring descriptive flexfields is an optional task.

Context

A descriptive flexfield can have only one context segment to provide context sensitivity. The same underlying database column can be used by different segments in different contexts.

For example, you can define a Dimensions context that uses the following attributes:

- ATTRIBUTE1 column for height
- ATTRIBUTE2 column for width
- ATTRIBUTE3 column for depth

You can also define a Measurements context that uses the same columns for other attributes:

- ATTRIBUTE1 column for weight
- ATTRIBUTE2 column for volume
- ATTRIBUTE3 column for density

Segments and Contexts

The following table lists the different types of descriptive flexfield **segments**.

<table>
<thead>
<tr>
<th>Segment Type</th>
<th>Run Time Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global segment</td>
<td>Always available</td>
</tr>
<tr>
<td>Context segment</td>
<td>Determines which context-sensitive segments are displayed</td>
</tr>
<tr>
<td>Context-sensitive segment</td>
<td>Displayed depending on the value of the context segment</td>
</tr>
</tbody>
</table>
The following figure displays a descriptive flexfield having one context segment called Category for which there are three values: Resistor, Battery, and Capacitor. Additionally, the descriptive flexfield comprises two global segments that appear in each context, and three context-sensitive segments that only appear in the specific context.

Application development determines the number of segments available for configuring. During implementation, configure the flexfield by determining the following:

- Attributes to add using the available segments
- Context values
- The combination of attributes in each context

Value Sets
For each global and context-sensitive segment, you configure the values permitted for the segment. Based on it, the values that end users enter are validated, including interdependent validation among the segments.

Protected Descriptive Flexfield Data
Application developers may mark some data configurations in a descriptive flexfield as protected, indicating that you can’t edit them.
Considerations for Planning Descriptive Flexfields

Once you have identified a flexfield to configure, plan the configuration in advance. Compile a list of the UI pages and other artifacts in your deployment that are affected by the configuration. Verify that you are provisioned with the roles needed to view and configure the flexfield. View the flexfield using the Highlight Flexfields command in the Administration menu while viewing the run time page where the flexfield appears. Plan how you will deploy the flexfield for test and production users. Review the tools and tasks available for managing flexfields for adding and editing flexfield segments.

Planning a descriptive flexfield can involve the following tasks:

1. Identify existing parameters.
2. Identify existing context values and whether the context value is derived.
3. Identify user-defined attributes and plan the descriptive flexfield segments, segment properties, and structure.
5. Plan initial values.
6. Plan attribute mapping to Oracle Business Intelligence objects.

Identify Existing Descriptive Flexfield Parameters

Some descriptive flexfields provide parameters that can be used to specify the initial value of a descriptive flexfield segment. The parameter is external reference data, such as a column value or a session variable. For example, if a flexfield has a user email parameter, you can configure the initial value for a customer email attribute to be derived from that parameter.

Review the list of available parameters in the Derivation Value field in the Create Segment page for a descriptive flexfield. If you decide to use one of the parameters to set an initial value, select that parameter from the Derivation Value drop-down list when you add the descriptive flexfield segment.

Evaluate Whether the Context Value Is Derived

The context value for a descriptive flexfield might have been preconfigured to be derived from an external reference. For example, if the context is Marriage Status, then the value might be derived from an attribute in the employee business object. When the context value is derived, you might need to take the derived values and their source into consideration in your plan.

To determine whether the context value is derived, access the Edit Descriptive Flexfield task to view the list of configured context values for the flexfield. The Derivation Value field in the Context Segment region displays a list of available parameters. If context values have been preconfigured, see Oracle Applications Cloud Help for product-specific information about the use of those values.

Plan the Segments, Segment Properties, and Structure

Identify the user-defined attributes you need for a business object to determine the segments of the descriptive flexfield. Determine the segment properties such as the prompt, display type, or initial value.

The structure of the descriptive flexfield is determined by its global, context, and context-sensitive segments. Plan a global segment that captures an attribute for every instance of the business object. Plan a context for segments that depend on a condition of situation applying to a particular instance of the business object. Plan context-sensitive segments to capture attributes that are relevant in the context.

There is only one context segment available for descriptive flexfields. If you have more than one group of user-defined attributes where you could use the context segment, you will have to pick one group over the others, based on your company's needs and priorities, and add the other user-defined attributes as global segments.
Plan Validation Rules
Define each segment's validation rules and check if value sets exist for those rules or you must create new ones. If you must create a value set, you can create it either before configuring the flexfield or while creating or editing a segment.

When determining a segment's validation rules, consider the following questions:

- What is the data type - character, date, date and time, or number?
- Does the segment require any validation beyond data type and maximum length?
- Should a character type value be restricted to digits, or are alphabetic characters allowed?
- Should alphabetic characters automatically be changed to uppercase?
- Should numeric values be zero-filled?
- How many digits can follow the radix separator of a numeric value? In base ten numeric systems, the radix separator is decimal point.
- Does the value need to fall within a range?
- Should the value be selected from a list of valid values? If so, consider the following questions:
  - Can you use an existing application table from which to obtain the list of valid values, or do you need to create a list?
  - If you are using an existing table, do you need to limit the list of values using a WHERE clause?
  - Does the list of valid values depend on the value in another flexfield segment?
  - Is the list of valid values a subset of another flexfield segment's list of values?

Plan Initial Values
For every segment, list the constant value or SQL statement, if any, to use for the initial value of the user-defined attribute.

Plan How Segments Map to Oracle Business Intelligence Objects
You can extend descriptive flexfields into Oracle Transactional Business Intelligence (OTBI) for ad hoc reporting purposes. Determine the descriptive flexfield segments to be made available for reporting, and select the BI Enabled check box accordingly on the Manage Descriptive Flexfields page. You must run a process to extend the BI enabled segments into OTBI. For more information about extending the BI enabled segments into OTBI, see the Flexfields chapter in the Oracle Applications Cloud Creating and Administering Analytics and Reports guide.

Depending on the reporting needs, you may map similar context-sensitive attributes from different contexts to the same attribute in OTBI. For example, there may be a segment tracking the Product Color attribute in different contexts of a context sensitive descriptive flexfield. You can use segment labels to map these context-sensitive attributes together by defining a segment label and updating the BI Label list accordingly.

Related Topics
• Overview of Transactional Business Intelligence Configuration of Descriptive Flexfields

Considerations for Managing Descriptive Flexfields
Configuring descriptive flexfields involves managing the available flexfields registered with your Oracle Applications Cloud database and configuring their flexfield-level properties, defining and managing descriptive flexfield contexts, and configuring global and context-sensitive segments.

Every descriptive flexfield is registered to include a context segment, which you may choose to use or not.

In general, configuring descriptive flexfields involves:

1. Creating segment labels for business intelligence enabled flexfields.
2. Configuring global segments by providing identity information, the initial default value, and the display properties.
3. Configuring the context segment by specifying the prompt, whether the context segment should be displayed, and whether a value is required.
4. Configuring contexts by specifying a context code, description, and name for each context value, and adding its context-sensitive segments, each of which is configured to include identifying information, the column assignment, the initial default value, and the display properties.

The following aspects are important in understanding descriptive flexfield management:

- Segments
- Adding segments to highlighted descriptive flexfields
- Usages
- Parameters
- Delimiters
- Initial Values
- Business Intelligence

Segments
You can assign sequence order numbers to global segments and to context-sensitive segments in each context. Segment display is always in a fixed order. You can't enter a number for a segment if that number is already in use for a different segment.

Value sets are optional for context segments and follow specific guidelines:

- The value set that you specify for a context segment consists of a set of context codes.
- Each context code corresponds to a context that's appropriate for the descriptive flexfield.
- The value set must be independent or table-validated.
- If table-validated, the WHERE clause must not use the VALUESET.value_set_code or SEGMENT.segment_code bind variables.
- The value set must be of data type Character with the maximum length of values being stored no larger than the context's column length.
- If you don't specify a value set for a context segment, the valid values for that context segment are derived from the context codes. The definition of each context segment specifies the set of context-sensitive segments that can be presented when that context code is selected by the end user.
- For reasons of data integrity, you can't 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're configured on application pages that use most of the same properties.

Adding Segments to Highlighted Descriptive Flexfields
When you highlight flexfields on a run time page and use an Add Segment icon button to create a segment, the segment code, name, description, table column, and sequence number are set automatically. If you use an Add Segment icon button to configure descriptive flexfield segments, you can't use an existing value set. Value sets are created automatically when you add the segments. You can enter the valid values, their descriptions, and the default value or specify the formatting constraints for the value set, such as minimum and maximum values.
Depending on display type, the value set you create using the **Add Segment** icon button is either an independent value set or a format-only value set. The following table shows which type of value set is created depending on the segment display component you select.

<table>
<thead>
<tr>
<th>Display Component</th>
<th>Value Set Created Using Add Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Box</td>
<td>Independent</td>
</tr>
<tr>
<td>Drop-down List</td>
<td>Independent</td>
</tr>
<tr>
<td>List of Values</td>
<td>Independent</td>
</tr>
<tr>
<td>Radio Button Group</td>
<td>Independent</td>
</tr>
<tr>
<td>Text Field With Search</td>
<td>Independent</td>
</tr>
<tr>
<td>Text box</td>
<td>Format Only</td>
</tr>
<tr>
<td>Text area</td>
<td>Format Only</td>
</tr>
<tr>
<td>Date/Time</td>
<td>Format Only</td>
</tr>
</tbody>
</table>

**Tip:** After you add a context value, refresh the page to see the new value.

**Usages**

Descriptive flexfield usages allow for the same definition to be applied to multiple entities or application tables, such as a USER table and a USER_HISTORY table. Descriptive flexfield tables define the placeholder entity where the flexfield segment values are stored once you have configured the descriptive flexfield. When you configure a flexfield, the configuration applies to all its usages.

**Parameters**

Some descriptive flexfields provide parameters, which are attributes of the same or related entity objects. Parameters are public arguments to a descriptive flexfield. Parameters provide outside values in descriptive flexfield validation. You use parameters to set the initial value or derivation value of an attribute from external reference data, such as a column value or a session variable, rather than from user input. Parameters can be referenced by the logic that derives the default segment value, and by table-validated value set WHERE clauses.

**Delimiters**

A segment delimiter or separator visually separates segment values when the flexfield is displayed as a string of concatenated segments.

**Initial Values**

The SQL statement defining an initial value must be a valid statement that returns only one row and a value of the correct type.
You can use two types of SQL statements:

- SQL statement with no binding. For example, select MIN(SALARY) from EMPLOYEES.
- SQL statement with bind variables. You can use these bind variables in the WHERE clause of the SQL statement.

  - `{SEGMENT.<segment_code>`: Identifies a segment in the same context.
  - `{PARAMETER.<parameter_code>`: Identifies a parameter.
  - `{CONTEXT.<context_code>; SEGMENT.<segment_code>`: Identifies a segment in a different context. The context must be in the same category or in an ancestor category, and it can't be a multiple-row context.
  - `{VALUESET.<value_set_code>`: Identifies the closest prior segment in the same context that's assigned to the specified value set.
  - `{FLEXFIELD.<internal_code>`: Identifies a flexfield.

**Business Intelligence**

Selecting a global, context, or context-sensitive segment’s BI Enabled check box specifies that the segment is available for use in Oracle Business Intelligence.

When the flexfield is imported into Oracle Business Intelligence, the label you selected from the BI Label drop-down list equalizes the segment with segments in other contexts, and maps the segment to the logical object represented by the label.

**Considerations for Enabling Descriptive Flexfield Segments for Business Intelligence**

A descriptive flexfield that is registered in the database as enabled for Oracle Business Intelligence (BI) includes a BI Enabled setting for each of its segments. When a global, context, or context-sensitive segment is BI-enabled, it is available for use in Oracle Business Intelligence.

The following aspects are important in understanding BI-enabled flexfield segments:

- Flattening business components to use BI-enabled segments in Oracle BI
- Equalizing segments to prevent duplication and complexity in the flattened component
- Mapping attributes of flattened business components to logical objects in Oracle BI
- Managing the labels that map segments to logical objects in Oracle BI

After you deploy a business intelligence-enabled flexfield, use the Import Oracle Fusion Data Extensions for Transactional Business Intelligence process to import the flexfield changes into the Oracle Business Intelligence repository. Users can make use of the newly-generated attributes in business intelligence applications. For example, a user can generate a report that includes attributes added by the descriptive flexfield. For additional information about logical objects and import, refer to the Oracle Transactional Business Intelligence Administrator’s Guide.

**Flattening**

When you deploy a business intelligence-enabled descriptive flexfield, the deployment process generates an additional set of flattened Application Development Framework (ADF) business components in addition to the usual ADF business components and ADF faces run time artifacts that are generated during deployment. The flattened business components include attributes for business intelligence-enabled segments only. Flattening means each user-defined column in each context shows up as an attribute in an Oracle Business Intelligence folder.
Flattened components include one attribute for the BI-enabled context-segment, and one attribute for each business intelligence-enabled global segment. For BI-enabled context-sensitive segments, consider the following:

- If you assigned a label to the segment, the flattened components include an additional single attribute representing segments with that label.
- If you didn't assign a label, the flattened components include a discrete attribute for each BI-enabled context-sensitive segment in each context.

Mapping to Logical Objects in Business Intelligence

You can simplify reporting by representing similar segments as a single logical object in Business Intelligence.

If you assign a label to any set of context-sensitive segments that serve the same purpose in different contexts, you can consolidate or equalize the segments into a single attribute. This prevents duplication and the extra workload and complexity that result from the flattening process. For example, a United States context might have a Passport segment and a Canada context might have Visa segment. If you assign the NationalID segment label to both the Passport and Visa segments, they are equalized into the same NationalID attribute in the flattened business component.

Non-labeled context-sensitive segments aren’t equalized across context values, so the flattened components include a separate attribute for each context-sensitive segment for each context value. It may not be possible to equalize similarly labeled segments if they have incompatible data types or value set types.

Assign a label to a global segment, context segment, or context-sensitive segment to map the corresponding attribute in the flattened components to a logical object in Oracle Business Intelligence. Using labels to map segments to BI logical objects minimizes the steps for importing the flexfield into Oracle Business Intelligence.

Note: Assigning a label to a context-sensitive segment serves to equalize the attribute across contexts, as well as map the equalized attribute to business intelligence.

Managing Labels

You may assign a predefined label (if available) to segments or create new labels for assignment, as needed. Specify a code, name, and description to identify each label. In the BI Object Name field, enter the name of the logical object in Oracle Business Intelligence to which the segment label should map during import. Specifying the BI logical object minimizes the steps for importing the flexfield into Oracle Business Intelligence and helps to equalize context-sensitive segments across contexts.

If no labels are assigned to a BI-enabled segment, or the BI Object Name on the assigned label doesn’t exist in business intelligence, you must manually map the segment to the desired logical object when importing into Oracle Business Intelligence.

In addition, context-sensitive segments without labels cannot be equalized across context values. The flattened components include a separate attribute for each non-labeled context-sensitive segment in each context.

Importing to Oracle Business Intelligence Repository

After you deploy a business intelligence-enabled flexfield, import the flexfield changes into the Oracle Business Intelligence repository to make use of the newly flattened business components in business intelligence and then propagate the flexfield object changes. When you import the metadata into the Oracle Business Intelligence repository, you must do so as the FUSION_APPS_BI_APPID user.

To import flexfield changes into the Oracle Business Intelligence repository in Oracle Cloud implementations, run the Import Oracle Fusion Data Extensions for Transactional Business Intelligence process. For additional information about import, refer to the Oracle Transactional Business Intelligence Administrator’s Guide.
Manage Extensible Flexfields

Overview of Extensible Flexfields

Extensible flexfields are like descriptive flexfields, with some additional features.

- You can add as many context-sensitive segments to the flexfield as you need. You aren't restricted by the number of columns predefined and registered for the flexfield.
- You can configure a one-to-many relationship between the entity and its extended attribute rows.
  - A row of data can have multiple contexts associated with it.
  - A row of data can have multiple occurrences of the same context.
- You can configure attributes in groups to form a context so that 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.
- Application development has registered some extensible flexfields to support view and edit privileges. For such flexfields, you can specify view and edit privileges at the context level to control who sees the attributes and who can change the attributes' 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 doesn't differ by context value. An extensible flexfield describes an application entity, with the run time ability to expand the database that implementation consultants can use to define the data structure that appears in the application. Extensible flexfields support one-to-many relationships between the entity and the extended attribute rows. To get a list of predefined extensible flexfields, use the Manage Extensible Flexfields task in the Setup and Maintenance work area.

The following aspects are important in understanding extensible flexfields:

- Usages
- Categories
- Pages
- Security
- Protected Extensible Flexfield Data

Usages

Similar to the descriptive flexfields, you can define multiple usages for an extensible flexfield, which enables several application tables to share the same flexfield.

For example, a flexfield for shipping options can be used by both a Supplier table and a Buyer table. In addition, you can associate a context with one, some, or all of the flexfield's usages. Thus, with the shipping information example, you can
associate a warehouse context with the Supplier usage, a delivery location context with the Buyer usage, and a ship-via context with all usages.

Usages include security information for applying no security to user access or enforcing view and edit privileges. Some product-specific extensible flexfields have specialized usage fields beyond those for security.

Categories
You can configure multiple extensible flexfield contexts and group the contexts into categories. All extensible flexfields have at least one category. For some extensible flexfields, you can configure a hierarchy of categories. A child category in the hierarchy can inherit contexts from its parent category.

You can define categories for extensible flexfields, and you can associate any combination of contexts with a given category.

For example, the Electronics and Computers category hierarchy might include a Home Entertainment category, which in turn might include an Audio category and a TV category, and so on. The Home Entertainment product might have contexts that specify voltage, dimensions, inputs and outputs. Contexts are reusable within a given extensible flexfield. For example, the dimensions context could be assigned to any category that needs to include dimensional information.

Pages
Extensible flexfields let you combine contexts into groups known as pages, which serve to connect the contexts so they will always be presented together in the application user interface.

Each application page corresponds to one extensible flexfield category, with a separate region of the page for each associated context.

Security
When you configure a flexfield, you set the privileges for a context at the usage level by selecting actions for the view and edit privileges of a context usage.

When an end user performs a search, the user interface displays only the attribute values of the contexts for which the user has view privileges. The user can perform a search using all attributes for all contexts, regardless of view privileges.

If end users access a context through a web service, an exception is thrown if they perform an action for which they don't have privileges.

All extensible flexfields have a base data security resource. Some data security resources for extensible flexfields are preconfigured with actions that you can use to specify access privileges. If no action is preconfigured, a security administrator can create actions and policies to support access control on the extensible flexfield attributes.

Some extensible flexfields have a translatable option; these flexfields also have a translation data security resource.

Protected Extensible Flexfield Data
Application developers may mark some data configurations in an extensible flexfield as protected, indicating that you can't edit them.

If an extensible flexfield is partially protected, then you can't edit the protected portions of the flexfield's configuration. For example:

- If an extensible flexfield context is protected, you can't edit its:
  - Context details
  - Context segments
• Context usages

  • If an extensible flexfield page is protected, you can’t:
    o Edit the page details or delete the page
    o Edit the contexts associated with the page

  **Note:**
  • There is no restriction on page references to protected contexts. The pages you create may contain any context, whether protected or not.
  • There is a restriction on category references to protected contexts. If a context is protected, you can’t add it to or delete it from any category.

**Related Topics**
  • Update Existing Setup Data

**Considerations for Planning Extensible Flexfields**

Once you have identified a flexfield, plan its configuration in advance. Compile a list of the UI pages and other artifacts in your deployment that are affected by the configuration. Verify that you are provisioned with the roles required to view and configure the flexfield. View the flexfield using the Highlight Flexfields option in the Administration menu while viewing the run time page where the flexfield appears. Plan how to deploy the flexfield for test and production users. Review the tools and tasks available for managing flexfields for adding and editing flexfield segments.

Planning an extensible flexfield involves:

1. Identifying the following:
   o A hierarchical structure of categories
   o Existing context values
   o User-defined attributes, the relevant extensible flexfield segments, segment properties, and the structure

2. Planning the following:
   o Validation rules
   o Initial values
   o Security
   o Attribute mapping to Oracle Business Intelligence objects.

**Category Hierarchy Structure**

Existing category hierarchy structures provide the framework for planning what segments to add to an extensible flexfield as user-defined attributes of an entity. Some applications provide user interfaces to create and manage an extensible flexfield’s category hierarchy.

**Contexts and Existing Context Values**

If related attributes can be grouped together, plan adding the attributes as a context of segments, and plan the order in which the attributes should appear. Some extensible flexfields have preconfigured context values. Region headers displayed in the user interface page or pages that contain the flexfield segments identify existing contexts. Using the Manage Extensible Flexfields task, find and open the flexfield for editing to view the list of configured context values.

See product-specific information for guidance in using preconfigured context values.
Plan the Segments and Segment Properties
List all the user-defined attributes that you want to add as extensible flexfield segments. For each segment, define properties, including the indexed property.

Plan Validation Rules
Define each segment's validation rules and check if value sets exist for those rules or you must create. If you must create a value set, you can create it either before you configure the flexfield or at the same time that you create or edit a segment.

When determining a segment's validation rules, consider the following questions:

- What is the data type: character, date, date and time, or number?
- Does the segment require any validation beyond data type and maximum length?
- Should a character type value be restricted to digits, or are alphabetic characters permitted?
- Should alphabetic characters automatically be changed to uppercase?
- Should numeric values be zero-filled?
- How many digits can follow the radix separator of a numeric value? In base ten numeric systems, the radix separator is a decimal point.
- Should the value be within a range?
- Should the value be selected from a list of valid values? If yes, consider the following questions:
  - Can you use an existing application table from which to obtain the list of valid values, or do you have to create a list?
  - If you are using an existing table, do you have to limit the list of values using a WHERE clause?
  - Does the list of valid values depend on the value in another flexfield segment?
  - Is the list of valid values a subset of another flexfield segment's list of values?

Plan Initial Values
For every segment, list the constant value or SQL statement, if any, to use for the initial value of the user-defined attribute.

Plan Security
Determine what privileges to set for view and edit access to context attributes, such as providing all users with view access but only managers with edit access.

If your security restrictions apply to several contexts, you can create generic actions. At a minimum, create the generic actions for the base data security resource. If the flexfield has a translatable option and you plan to use translatable contexts, then also create the generic actions for the translation data security resource. For example, the Item flexfield supports the translatable option and has a data security resource ITEM_EFF_VL in addition to the base data security resource ITEM_EFF_B. Then, create actions for both data security resources, such as EDIT_NONTRANS_ATRTRS for ITEM_EFF_B and EDIT_TRANS_ATRTRS for ITEM_EFF_VL.

If your security restrictions are more fine-grained, such as having to secure each context with a different privilege, then you can create more fine-grained actions.

Plan Which Segments Map to Oracle Business Intelligence Objects
If an extensible flexfield has been enabled for Oracle Business Intelligence, you can make the attributes available for use in Oracle Business Intelligence Applications.
Considerations for Managing Extensible Flexfields

Configuring extensible flexfields involves managing the available flexfields registered with your application database. The following sequence describes how to configure extensible flexfields:

1. Configuring contexts by creating each context segment and the context-sensitive segments for each context segment, and providing the following for each segments:
   a. Identifying information
   b. Column assignment
   c. Initial default value
   d. Display properties
2. Configuring context usages and usage security by selecting actions to which users should have access:
   o View
   o Edit
   o None, if no special privileges should be enforced.
3. Configuring categories and category details.
4. Associating contexts with a category.
5. Creating logical pages for a category.

The following aspects are important in understanding extensible flexfield management:

- Contexts and pages
- Categories
- Initial values
- Adding segments to highlighted extensible flexfields
- Indexed segments
- Security
- Deployment

Contexts and Pages

Each context is displayed to end users as a region containing its context-sensitive segments. You can specify instruction help text to display instructions that explain how to use the region and its attributes to end users. Instruction help text is displayed at the beginning of the context region. A context can be defined as single row or multi-row. Single row contexts are the same as descriptive flexfields contexts. A single row context has only one set of context-sensitive segments. A multi-row context enables you to associate multiple sets of values with the same object instance.

For example, for a BOOK table, you could create a multi-row context named chapters that contains a segment for chapter and a segment for number of pages. Multiple chapters can then be associated with each book in the BOOK table.

For contexts that store multiple rows, you can uniquely identify each row by having the values in each row form a unique key.

If flexfield has a category hierarchy, then you can leverage the hierarchy to reuse contexts for similar entities, such as similar items in a product catalog.

Set the context to translatable so that free-form text entered by end users is stored in the language of the user’s locale, and different translations of that text can be stored in other languages. Segments in the translated contexts should use format-only value sets for storing free-form, user-entered text.
Set the context security to give an end user view or edit access to a context. The context's task flow and region appear in the user interface only for users with view access. With edit access, an end user can edit the context's attribute values. With no action specified for a usage, no special privileges are enforced through the context's configuration.

Define logical pages to group contexts together in the user interface. For a given category, you may create one or more logical pages. You may add one or more of the category's associated contexts to each of the category's logical pages.

You can specify:

- The sequence of the contexts within each page.
- The sequence in which the logical pages appear.
- Instruction help text to display instructions that explain how to use the page to end users. Instruction help text is displayed at the beginning of the logical page, preceding all of its context regions.

Categories
A category is a grouping of related data items that can be considered to belong together. You can associate any combination of contexts with a given category. Extensible flexfields with more than 30 categories must be deployed as a background process.

A category hierarchy logically organizes a set of categories. For example, the Electronics and Computers category hierarchy might include a Computer category and a Home Entertainment category, which in turn might include an Audio category and a TV category, and so on.

A category can be a child or sibling of an existing category. The hierarchy can be as simple or as complex as desired, with any combination of zero or more sibling categories and zero or more child categories. If no category is defined, the data items are grouped in 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. Then, the Dimensions context could be assigned to any category that needs to include dimensional information.

If a hierarchy includes child categories, each child category inherits the contexts from its parent category; for example, the Home Entertainment category inherits Voltage and Dimensions from the Electronics and Computers category.

Each extensible flexfield is associated with a particular category hierarchy. Consider category hierarchies to be defining framework for extensible flexfields and their contexts. A category hierarchy specifies which contexts are valid for each category.

An extensible flexfield can include multiple contexts which you define to support a given category. These contexts can be suitable for various purposes, but within a particular category, some contexts might be considered to be related to, or dependent on, each other. You can combine these contexts into groups known as logical pages, and determine the sequence in which the pages appear. This serves to connect the contexts so they will always be presented together and in a particular order in the application user interface.

For example, the Home Entertainment category might have an Electrical Specifications page that contains the Voltage, Inputs and Outputs contexts, and a Physical Specifications page that contains the Dimensions and Form Factor contexts.

Initial Values
The SQL statement defining an initial value must be a valid statement that returns only one row and a value of the correct type.

You can use two types of SQL statements:

- SQL statement with no binding. For example, select MIN(SALARY) from EMPLOYEES.
• SQL statement with bind variables. You can use these bind variables in the WHERE clause of the SQL statement.

  o `{SEGMENT.<segment_code>`: Identifies a segment in the same context.

  o `{PARAMETER.<parameter_code>`: Identifies a parameter.

  o `{CONTEXT.<context_code>;SEGMENT.<segment_code>`: Identifies a segment in a different context. The context must be in the same category or in an ancestor category, and it can't be a multiple-row context.

  o `{VALUESET.<value_set_code>`: Identifies the closest prior segment in the same context that's assigned to the specified value set.

  o `{FLEXFIELD.<internal_code>`: Identifies a flexfield.

Adding Segments to Highlighted Extensible Flexfields

When you highlight flexfields on a run time page and use an Add Segment icon button to create a segment, the segment code, name, description, table column, and sequence number are set automatically. If you use an Add Segment icon button to configure extensible flexfield segments, you can't use an existing value set. Value sets are created automatically when you add segments. You can enter the valid values, their descriptions, and the default value or specify the formatting constraints for the value set, such as minimum and maximum values.

Depending on display type, the value set you create with the Add Segment icon button is either an independent value set or a format-only value set. The following table shows which type of value set is created depending on the segment display component you select.

<table>
<thead>
<tr>
<th>Display Component</th>
<th>Value Set Created Using Add Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Box</td>
<td>Independent</td>
</tr>
<tr>
<td>Drop-down List</td>
<td>Independent</td>
</tr>
<tr>
<td>List of Values</td>
<td>Independent</td>
</tr>
<tr>
<td>Radio Button Group</td>
<td>Independent</td>
</tr>
<tr>
<td>Text Field With Search</td>
<td>Independent</td>
</tr>
<tr>
<td>Text box</td>
<td>Format Only</td>
</tr>
<tr>
<td>Text area</td>
<td>Format Only</td>
</tr>
<tr>
<td>Rich Text Editor</td>
<td>Format Only</td>
</tr>
<tr>
<td>Date/Time</td>
<td>Format Only</td>
</tr>
</tbody>
</table>
Tip: After you add a context value, refresh the page to see the new value.

Indexed Segments
You can designate an extensible flexfield segment as indexed so that it's one of the selectively required attributes a user can use in an attribute search. If you indicate in the Manage Extensible Flexfield UI page that a segment should be indexed, the column representing the segment must be added to the database index. Commonly, a database administrator (DBA) adds columns to the database index.

When an extensible flexfield with indexed segments is deployed, search task flows are generated along with the other flexfield artifacts and specify the indexed attributes as selectively required. In the deployed extensible flexfield's search task flow, an end user must specify at least one of the indexed attributes in the search criteria. This prevents non-selective searches, which could cause performance issues.

For example, if you index the memory and processor attributes and ensure that the corresponding columns in the database are indexed, a user can search an item catalog for computers by entering processor or memory or both as a search criteria. No search is performed if an end user enters an attribute that isn't indexed as a search criterion.

Security
An extensible flexfield's base data security resource typically has a name with an _B suffix. The translation data security resource is a view of a translation table that typically has a name with an _VL suffix.

If a flexfield supports the translatable option and has a translation data security resource, make sure that you create the action for the appropriate data security resource.

- If you create a context-specific action for a nontranslatable context, add it to the base data security resource.
- If you create a context-specific action for a translatable context, add it to the translation data security resource.

Deployment
You can only deploy extensible flexfields using the Manage Extensible Flexfields task. You can deploy extensible flexfields offline as a background process and continue working in the session without having to wait for the deployment to complete. You can queue up several extensible flexfields and deploy as a background process. The flexfields are deployed, one at a time, in the order that you deploy them to the queue. You must deploy extensible flexfields with more than 30 categories as a background process.

You can remove an extensible flexfield from the deployment queue with the Cancel Background Deployment command. When an extensible flexfield is deployed in a background process, its offline status indicates that the flexfield is in a background deployment process. A flexfield's offline status is cleared and it's deployment status updated when the background deployment process has completed.

Note: The Offline Status column refreshes when you perform a new search in the Manage Extensible Flexfields task.

Considerations for Enabling Extensible Flexfield Segments for Business Intelligence
An extensible 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's available for use in Oracle Business Intelligence.

The following aspects are important in understanding BI-enabled extensible flexfield segments.

- Flattening business components to use BI-enabled segments in Oracle BI
- Mapping attributes of flattened business components to logical objects in Oracle BI

After you deploy a business intelligence-enabled flexfield, use the Import Oracle Fusion Data Extensions for Transactional Business Intelligence process to import the flexfield changes into the Oracle Business Intelligence
repository. Users can make use of the newly-generated attributes in business intelligence applications. For additional information about logical objects and import, refer to the Oracle Transactional Business Intelligence Administrator's Guide.

Flattening
When you deploy a business intelligence-enabled extensible flexfield, the deployment process generates an additional set of flattened business components for use in business intelligence. The flattened business components include attributes for business intelligence-enabled segment instances only.

If you assigned a label to a segment, the flattened components include a single attribute representing all segment instances with that label. If you didn't assign a label, the flattened components include a discrete attribute for each BI-enabled segment instance in each structure.

Importing to Oracle Business Intelligence Repository
After you deploy a business intelligence-enabled flexfield, import the flexfield changes into the Oracle Business Intelligence repository to make use of the newly flattened business components in business intelligence and then propagate the flexfield object changes. When you import the metadata into the Oracle Business Intelligence repository, you must do so as the FUSION_APPS_BI_APPID user. To import flexfield changes into the Oracle Business Intelligence repository in Oracle Cloud implementations, run the Import Oracle Fusion Data Extensions for Transactional Business Intelligence process. For additional information about import, refer to the Oracle Transactional Business Intelligence Administrator's Guide.

**Tip:** When you import a flexfield into the Oracle Business Intelligence repository, you see both `<name>` and `<name>_c` attributes for each segment, along with some other optional attributes. The `<name>` attribute contains the value. The `<name>_c` attribute contains the code of the value set that the value comes from, and is used for linking to the value dimension. You must import both attributes.

Considerations for Managing Extensible Flexfield Categories
Categories are a way of extending the number of context-sensitive segments for a flexfield beyond the columns reserved for flexfield segments.

For example, an Items extensible flexfield has a category for each item and each category can have one or more contexts. The laptop item belongs to the Computers category. Since extensible flexfields are mapped to separate extension tables, not just to columns as with descriptive flexfields, the thirty reserved columns on the extensible flexfield table let you define up to thirty context-sensitive segments for each context.

If you add a Dimensions context to the Computers category, thirty segments are available. But if you need to add more than thirty attributes, create another context and associate it to the same category. You could now add an Electronics Attributes context to the same Computers category in which you create another thirty segments. You can continue creating more contexts and adding them to the Computers category. In this way your laptop computer item can be extended with as many attributes as you need, because it is mapped to a category and you can keep adding contexts to that category.

A descriptive flexfield on an items table with thirty columns reserved for segments can only have a single context. Once you configure the columns for that one context, you cannot create any more segments.

Predefined and Preconfigured Categories
How you structure the flexfield configuration depends on how categories are defined for the flexfield. If the extensible flexfield is preconfigured with one category, associate all your contexts and pages with that category. If a product-specific extensible flexfield is preconfigured with several categories, associate your contexts and pages with those categories. If the extensible flexfields provide user interfaces for configuring multiple categories, associate a context with more than one category using inheritance.
Some products provide an activity or task for creating and maintaining categories for an extensible flexfield. See product-specific information to determine if you can create categories for the flexfield.

You can view a flexfield's category hierarchies by using either the Highlight Flexfields feature or the Manage Extensible Flexfields task to find and open the flexfield for editing.

Disabling Categories
While configuring an extensible flexfield, you can disable a category. The Enabled column in the Category table of the Edit Extensible Flexfield page, indicates which categories are enabled.

**Note:** When you deploy an extensible flexfield that has a disabled category, that category and its descendant categories aren't deployed. Contexts and their segments are deployed only if they belong to at least one enabled category.

Contexts
Group similar attributes into contexts. The group is displayed together in a region. The region's header is the context value.

If a category hierarchy exists for the flexfield, then you can leverage the hierarchy to reuse contexts for similar entities, such as similar items in a product catalog.

The following figure shows the Item Extended Attributes flexfield, which uses the category hierarchy feature to reuse contexts. The flexfield's Electronics and Computers category contains contexts for compliance and certification, voltage, and materials and substances. The TV and Video subcategory and the Computer Products subcategory inherit the Electronics and Computer contexts in addition to having their own contexts. The Materials and Substances context belongs to both the Electronics and Computer Products category and the Tools, Auto, and Industrial Products category.

The following table shows an example of category hierarchy for an extensible flexfield. To store voltage information for all electronic and computer items, associate a Voltage context with the Electronics and Computers category. Both the TV and Video subcategory and the Computers subcategory then inherit the Voltage context from the parent Electronics and Computers category.
### Example of Configuring an Item Extended Attributes Flexfield

The Item Extended Attributes flexfield provides segments for extending the Item business object. In the Manage Extensible Flexfields task, configure your product business object to include a technical specifications logical page on the user interface for the Electronics and Computers category of items.

In this example, your configuration of this flexfield groups attributes into the following contexts:

- Materials and Substances
- Compliance and Certification
- Voltage

### Scenario

The following list shows an example plan for computer attributes for the Item Extended Attributes flexfield. In this example, the Electronics Information page is inherited from the parent Electronics and Computers category.

- Page: Electronics Information
  - Context: Compliance and Certification, single row
    - ISO 14001 (International Organization for Standardization for an Environmental Management System)
    - ENERGY STAR (energy efficiency guidelines)
    - ROHS (Restriction of the use of certain hazardous substances in electrical and electronic equipment)
  - Context: Voltage, single row
    - Minimum voltage
    - Maximum voltage
    - Current type
  - Context: Materials and Substances, multiple rows
    - Material

---

<table>
<thead>
<tr>
<th>Display Name</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronics and Computers</td>
<td>PROD_ELECTRONICS</td>
<td>Electronics and Computers</td>
</tr>
<tr>
<td>• TV and Video</td>
<td>PROD_TV_VIDEO</td>
<td>Television and Video</td>
</tr>
<tr>
<td>• Computers</td>
<td>PROD_COMPUTERS</td>
<td>Computers</td>
</tr>
<tr>
<td>Office Products and Supplies</td>
<td>PROD_OFFICE_PRODUCTS_SUPPLIES</td>
<td>Office Products and Supplies</td>
</tr>
<tr>
<td>Tools, Auto, and Industrial</td>
<td>PROD_TOOLS_AUTO_INDUSTRIAL</td>
<td>Tools, Automotive, and Industrial</td>
</tr>
<tr>
<td>Sports and Outdoors</td>
<td>PROD_SPORTS_OUTDOORS</td>
<td>Sports and Outdoors</td>
</tr>
</tbody>
</table>
- Contain recyclate
- Percent unit mass

- Page: Computer Information
  - Context: Processor Specifications, single row
    - Manufacturer
    - CPU type
    - Processor interface
    - Processor class
    - Processor speed
    - Cores

The following table summarizes key decisions for this scenario.

<table>
<thead>
<tr>
<th>Decisions to Consider</th>
<th>In This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Which extensible flexfield is available for configuring a hierarchy of categories?</td>
<td>Item Extended Attributes flexfield</td>
</tr>
</tbody>
</table>

Collecting Technical Specifications

Your product inventory pages for electronics and computers require a technical specifications page. Your product inventory pages for furniture require a furniture specifications page and an assembly instructions page. Items in both the electronics and computer category, and in the furniture category, share attributes for specifying materials and substances.

The following figure shows a technical specifications logical page in the user interface for the Electronics and Computers category. It contains attributes in the context of Recovery and Recycling, Compliance and Certification,
Operating Conditions, and Materials and Substances. The Materials and Substances context is configured for multiple rows. Your users can select all the materials and substances required to make a single product.

Analysis

Use logical pages to determine how the contexts appear on the user interface. Use a context to store all the materials and substances required to make a single product. You can configure a context to store multiple rows per entity. The multiple rows are displayed in a table, like the Materials and Substances context.

The Technical Specifications logical page contains the attributes for the four contexts.

- Recovery and Recycling
- Compliance and Certification
- Operating Conditions
- Materials and Substances
The following figure is an example, where the Furniture category is configured to include a Furniture Specifications logical page and an Assembly Instructions logical page. The two categories (Electronics and Computers and Furniture) share the Materials and Substances context.

Configure Security for the Item Flexfield Configuration

The following table shows an example of data security policies for the Item flexfield.

<table>
<thead>
<tr>
<th>Data Security Resource</th>
<th>Policy</th>
<th>Role</th>
<th>Action</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITEM_EFF_B</td>
<td>A</td>
<td>VOLTAGE_SPEC</td>
<td>edit_nontrans_voltage_ctx</td>
<td>All values</td>
</tr>
<tr>
<td>ITEM_EFF_VL</td>
<td>B</td>
<td>COMPLIANCE_SPEC</td>
<td>edit_trans_compliance_ctx</td>
<td>All values</td>
</tr>
<tr>
<td>ITEM_EFF_VL</td>
<td>C</td>
<td>COMPUTER_SPEC</td>
<td>edit_trans_attrs</td>
<td>ComputerCategoryFilter</td>
</tr>
<tr>
<td>ITEM_EFF_VL</td>
<td>D</td>
<td>TELEVISION_SPEC</td>
<td>edit_trans_attrs</td>
<td>TVCategoryFilter</td>
</tr>
</tbody>
</table>

The following table shows the privileges for the three flexfield contexts.
In this example, anyone can view the contexts' attributes, but the edit privileges are restricted as follows:

- **Voltage**: Only voltage specialists can edit this value.
- **Compliance and Certification**: Only compliance specialists can edit this value.
- **Materials and Substances**: Only computer specialists can edit these attributes for items in the computer category. Only television specialists can edit these attributes for items in the TV category.

To sum up, in this entire example, the Materials and Substances context is secured by a generic action with a condition applied to restrict access by category. Voltage and Compliance and Certification are secured by actions specific to each context.

**File Format for Importing Extensible Flexfields**

To import extensible flexfields into an application, you create separate text files for the value sets, contexts, and context segments. Then, you upload them to the Oracle WebCenter Content document repository. Each file must follow a specific format, as described here. After the files are placed in the document repository, you can import the values sets, contexts, and context segments into the application.

While creating the file, adhere to the following guidelines:

- Use a vertical bar or pipe ( | ) as a delimiter between fields for both header and value rows.
- Set the file encoding to UTF-8 without the Byte Order Mark (BOM) as per the Oracle WebCenter Content specification.

The following sections contain specific details about each file format.

**Prerequisite**

You must have worked with flexfields in Oracle Cloud applications.

**Value Sets**

To create a file containing value sets, include the headers listed in the following table:

<table>
<thead>
<tr>
<th>Header</th>
<th>Data Type</th>
<th>Data Size</th>
<th>Description</th>
<th>Required or Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueSetCode</td>
<td>String</td>
<td>60</td>
<td>This value identifies your value set across components.</td>
<td>Required</td>
</tr>
<tr>
<td>ModuleType</td>
<td>String</td>
<td>60</td>
<td>This value determines the module type of your value set.</td>
<td>Required</td>
</tr>
<tr>
<td>Header</td>
<td>Data Type</td>
<td>Data Size</td>
<td>Description</td>
<td>Required or Optional</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>ModuleKey</td>
<td>String</td>
<td>60</td>
<td>This value determines the specific module of your value set.</td>
<td>Required</td>
</tr>
<tr>
<td>ValidationType</td>
<td>String</td>
<td>30</td>
<td>This value determines your value set type. The values you can use are DEP, INDEP, FORMAT, SUBSET, and RELATED.</td>
<td>Required</td>
</tr>
<tr>
<td>ValueDataType</td>
<td>String</td>
<td>30</td>
<td>This value determines the data type that your value set uses. Supported data types are VARCHAR2, NUMBER, DATE, and TIMESTAMP.</td>
<td>Required</td>
</tr>
<tr>
<td>ValueSubType</td>
<td>String</td>
<td>30</td>
<td>This value determines the data subtype your value set uses. Supported data subtypes are TEXT, TIME_HM, NUMERIC, TIME_HMS, and TRANSLATED</td>
<td>Required for ValueDataType VARCHAR2</td>
</tr>
<tr>
<td>MaximumLength</td>
<td>Integer</td>
<td>-</td>
<td>This value determines the maximum length of values in your value set.</td>
<td>Required for ValueDataType VARCHAR2</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>240</td>
<td>This value gives your value set a description.</td>
<td>Optional</td>
</tr>
<tr>
<td>Precision</td>
<td>Integer</td>
<td>-</td>
<td>This value determines the number of digits in the number data you add in your value set.</td>
<td>Optional Use this header only for ValueDataType NUMBER</td>
</tr>
<tr>
<td>Scale</td>
<td>Integer</td>
<td>-</td>
<td>This value determines the number of digits after the decimal point in the number data you add in your value set.</td>
<td>Optional Use this header only for ValueDataType NUMBER</td>
</tr>
<tr>
<td>UppercaseOnlyFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, then your value</td>
<td>Optional</td>
</tr>
<tr>
<td>Header</td>
<td>Data Type</td>
<td>Data Size</td>
<td>Description</td>
<td>Required or Optional</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------</td>
<td>-----------</td>
<td>----------------------------------------------------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>set will accept only uppercase characters. If it's N, then both uppercase</td>
<td>Use this header only for ValueDataType</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>and lowercase characters are supported.</td>
<td>VARCHAR2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If this value is set to Y, then zeroes are added to the left of the text</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>you add in your value set, till the text length matches your MaximumLength.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If the value is N, no zeroes are added.</td>
<td></td>
</tr>
<tr>
<td>ZeroFillFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, then zeroes are added to the left of the text</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>you add in your value set, till the text length matches your MaximumLength.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If the value is N, no zeroes are added.</td>
<td></td>
</tr>
<tr>
<td>SecurityEnabledFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, you can add a data security resource to your</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>value set. If the value is N, no data security resource is added.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DataSecurityObjectName</td>
<td>String</td>
<td>60</td>
<td>This value selects the data security resource you want to add to your</td>
<td>Required if SecurityEnabledFlag is set to Y.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>value set. This value must match a data security resource in the application.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MinimumValue</td>
<td>String</td>
<td>150</td>
<td>This value specifies the minimum value for the values in your value set.</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaximumValue</td>
<td>String</td>
<td>150</td>
<td>This value specifies the maximum value for the values in your value set.</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IndependentValueSetCode</td>
<td>String</td>
<td>60</td>
<td>This value determines the independent value set your dependent or subset</td>
<td>Required for ValidationType DEP and SUBSET</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>value set is linked to.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Don't use this header for validation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>types other than DEP and SUBSET.</td>
</tr>
</tbody>
</table>
Here's a sample file that contains the header values at the beginning of the file, followed by line entries of the three value sets that are to be imported. For importing several value sets, add more line entries in a similar format.

```
ValueSetCode|ModuleType|ModuleKey|ValidationType|ValueDataType|ValueSubtype|MaximumLength|Description
VS_TEST_91|APPLICATION|FND|INDEP|VARCHAR2|TEXT|2|desc1
VS_TEST_92|APPLICATION|FND|INDEP|VARCHAR2|TEXT|3|
VS_TEST_93|APPLICATION|FND|INDEP|VARCHAR2|TEXT|3|desc3
```

Context
To create a file containing the contexts, include the headers in the following table:

<table>
<thead>
<tr>
<th>Header</th>
<th>Data Type</th>
<th>Data Size</th>
<th>Description</th>
<th>Required or Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplicationId</td>
<td>Long</td>
<td>-</td>
<td>This value determines the Application to which your flexfield and context belongs to.</td>
<td>Required</td>
</tr>
<tr>
<td>EFFCode</td>
<td>String</td>
<td>40</td>
<td>This value is the code of the extensible flexfield to which you're adding your context. This value should match the code of an extensible flexfield that's already in the application.</td>
<td>Required</td>
</tr>
<tr>
<td>ContextCode</td>
<td>String</td>
<td>80</td>
<td>This value is the code for your context.</td>
<td>Required</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>80</td>
<td>This value determines the display name of your context.</td>
<td>Required</td>
</tr>
<tr>
<td>MultirowFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, then your context will support multi-row behavior. If the value is N, only single row behavior is supported.</td>
<td>Required</td>
</tr>
<tr>
<td>EnabledFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, then your context is enabled. If the value is N, then your context is disabled.</td>
<td>Required</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>240</td>
<td>This value gives your context a description.</td>
<td>Optional</td>
</tr>
<tr>
<td>Header</td>
<td>Data Type</td>
<td>Data Size</td>
<td>Description</td>
<td>Required or Optional</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------</td>
<td>-----------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>TranslatableFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, then the segments in your context are translatable. If the value is N, they're not.</td>
<td>Optional</td>
</tr>
<tr>
<td>ContextIdentifier</td>
<td>String</td>
<td>30</td>
<td>This value determines the API name for your context. There are a set of conventions to be followed when naming APIs.</td>
<td>Optional</td>
</tr>
<tr>
<td>InstructionHelpText</td>
<td>String</td>
<td>400</td>
<td>This text gives the user instructions about how to use the context region.</td>
<td>Optional</td>
</tr>
<tr>
<td>FlexfieldUsageCode</td>
<td>String</td>
<td>30</td>
<td>This value determines the usage code for your context. The value should match a usage code in the application.</td>
<td>Required</td>
</tr>
<tr>
<td>ViewPrivilegeName</td>
<td>String</td>
<td>400</td>
<td>This value determines which privileges can view this context usage. The value should match a privilege in the application.</td>
<td>Optional</td>
</tr>
<tr>
<td>EditPrivilegeName</td>
<td>String</td>
<td>400</td>
<td>This value determines which privileges can edit this context usage. The value should match a privilege in the application.</td>
<td>Optional</td>
</tr>
<tr>
<td>CExtAttribute1</td>
<td>String</td>
<td>150</td>
<td>This header adds a user defined attribute to a context usage. You can use up to 5 headers of this type, CExtAttribute1 to CExtAttribute5 to add your user defined attributes.</td>
<td>Optional</td>
</tr>
<tr>
<td>CONTEXT_CExtAttribute1</td>
<td>String</td>
<td>150</td>
<td>This header adds a user defined attribute to a context.</td>
<td>Optional</td>
</tr>
</tbody>
</table>
Here's a sample file that contains the header values at the beginning and lists three contexts to be imported. For importing several contexts, add more entries in the same format.

```
<table>
<thead>
<tr>
<th>ApplicationId</th>
<th>EFFCode</th>
<th>ContextCode</th>
<th>Name</th>
<th>EnabledFlag</th>
<th>MultirowFlag</th>
<th>Description</th>
<th>FlexfieldUsageCode</th>
<th>ViewPrivilegeName</th>
<th>CExtAttribute1</th>
<th>CExtAttribute2</th>
<th>CExtAttribute3</th>
<th>CExtAttribute4</th>
<th>CExtAttribute5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>FLEX_SN_EFF1</td>
<td>OBJ_TEST_4</td>
<td>Object test 44</td>
<td>Y</td>
<td>N</td>
<td>desc 44 3363</td>
<td>FLEX_SN_EFF1_USAGE2</td>
<td>flex_sn_sitems_view</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>FLEX_SN_EFF1</td>
<td>OBJ_TEST_3</td>
<td>Object test 33</td>
<td>Y</td>
<td>N</td>
<td></td>
<td>FLEX_SN_EFF1_USAGE1</td>
<td>flex_sn_view</td>
<td></td>
<td></td>
<td>NEW_TEST11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>FLEX_SN_EFF1</td>
<td>OBJ_TEST_3</td>
<td>Object test 33</td>
<td>Y</td>
<td>N</td>
<td>new desc aug 14</td>
<td>FLEX_SN_EFF1_USAGE2</td>
<td>flex_sn_sitems_edit</td>
<td>CE1_TESTupd_aug15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

### Context Segment

To create a file containing context segments, include the headers in the following table:

<table>
<thead>
<tr>
<th>Header</th>
<th>Data Type</th>
<th>Data Size</th>
<th>Description</th>
<th>Required or Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplicationId</td>
<td>Long</td>
<td>-</td>
<td>This value determines the Application to which your flexfield, context, and segment belongs to.</td>
<td>Required</td>
</tr>
<tr>
<td>EFFCode</td>
<td>String</td>
<td>40</td>
<td>This value is the code of the extensible flexfield to which you're adding your segment. This value should match the code of an extensible flexfield that's already in the application.</td>
<td>Required</td>
</tr>
<tr>
<td>ContextCode</td>
<td>String</td>
<td>80</td>
<td>This value is the code for the context to which you're adding your segment. This value should match a context code that's in the application.</td>
<td>Required</td>
</tr>
<tr>
<td>SegmentCode</td>
<td>String</td>
<td>30</td>
<td>This value is the code for your segment.</td>
<td>Required</td>
</tr>
<tr>
<td>Header</td>
<td>Data Type</td>
<td>Data Size</td>
<td>Description</td>
<td>Required or Optional</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>60</td>
<td>This is the backend name for your segment.</td>
<td>Required</td>
</tr>
<tr>
<td>ColumnName</td>
<td>String</td>
<td>30</td>
<td>This value determines the table column your segment uses to save data. This value must match a column that's in the application.</td>
<td>Required</td>
</tr>
<tr>
<td>ValueSetCode</td>
<td>String</td>
<td>60</td>
<td>This value is the code for the value set you want to use in your segment. This value must match a value set code in the application.</td>
<td>Required</td>
</tr>
<tr>
<td>DisplayType</td>
<td>String</td>
<td>30</td>
<td>This value determines the display type of the segment. Valid values for this attribute are TEXT_BOX, TEXT_AREA, RICH_TEXT_EDITOR HIDDEN, LOV, POP_UP_LIST, DROP_DOWN_LIST, RADIO_BUTTON_GROUP, STATIC_URL, CHECKBOX, COLOR, DATE_TIME</td>
<td>Required</td>
</tr>
<tr>
<td>Prompt</td>
<td>String</td>
<td>80</td>
<td>This value determines the display name of the segment.</td>
<td>Required</td>
</tr>
<tr>
<td>ShortPrompt</td>
<td>String</td>
<td>80</td>
<td>This value determines the shortened display name of the segment.</td>
<td>Required</td>
</tr>
<tr>
<td>EnabledFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, then your segment is enabled. If the value is N, then it's disabled.</td>
<td>Required</td>
</tr>
<tr>
<td>RequiredFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, user input for this segment is mandatory during run time. If the</td>
<td>Required</td>
</tr>
<tr>
<td>Header</td>
<td>Data Type</td>
<td>Data Size</td>
<td>Description</td>
<td>Required or Optional</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Header</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Type</td>
<td>String</td>
<td>1</td>
<td>value is N, user input is optional.</td>
<td></td>
</tr>
<tr>
<td>Data Size</td>
<td>String</td>
<td>240</td>
<td>This value gives your segment a description.</td>
<td>Optional</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>20</td>
<td>This value determines the unit for the data you add in your segment.</td>
<td>Optional</td>
</tr>
<tr>
<td>Required or Optional</td>
<td>String</td>
<td>30</td>
<td>This value determines the type of default value your segment takes if the user doesn't enter any data in your segment. The types are CONSTANT, SQL, and GROOVY_EXPRESSION</td>
<td>Optional</td>
</tr>
<tr>
<td>Default Value</td>
<td>String</td>
<td>4000</td>
<td>This value determines the default value your segment takes if the user doesn't enter any data.</td>
<td>Optional</td>
</tr>
<tr>
<td>Display Width</td>
<td>Integer</td>
<td>-</td>
<td>This value determines the display width of your segment, in terms of maximum number of characters displayed in a line.</td>
<td>Optional</td>
</tr>
<tr>
<td>Header</td>
<td>Data Type</td>
<td>Data Size</td>
<td>Description</td>
<td>Required or Optional</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>DisplayHeight</td>
<td>Integer</td>
<td>-</td>
<td>This value determines the display height of your segment, in terms of maximum number of lines displayed in the segment.</td>
<td>Optional</td>
</tr>
<tr>
<td>CheckboxCheckedValue</td>
<td>String</td>
<td>30</td>
<td>This value determines the value of a checked check box in your segment.</td>
<td>Required for display type CHECKBOX.</td>
</tr>
<tr>
<td>CheckboxUncheckedValue</td>
<td>String</td>
<td>30</td>
<td>This value determines the value of an unchecked check box in your segment.</td>
<td>Required for display type CHECKBOX.</td>
</tr>
<tr>
<td>RangeType</td>
<td>String</td>
<td>30</td>
<td>This value sets two fields as two ends of a range of values. Set value as LOW or HIGH for your low end and high end fields respectively.</td>
<td>Optional</td>
</tr>
<tr>
<td>BIEnabledFlag</td>
<td>String</td>
<td>1</td>
<td>This value determines whether your segment is BI enabled. Y means it's, N means it's not.</td>
<td>Optional</td>
</tr>
<tr>
<td>MultirowUniqueKeyFlag</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, the segment is marked as the unique key for a multirow context. Default value for this header is N. At least 1 segment in a multirow context must be marked as the unique key.</td>
<td>Required for multi-row contexts</td>
</tr>
<tr>
<td>ShowValueDescription</td>
<td>String</td>
<td>1</td>
<td>If this value is set to Y, the value set description for the value set associated to the segment is displayed. If the value is N, it's not displayed.</td>
<td>Optional</td>
</tr>
<tr>
<td>SegmentIdentifier</td>
<td>String</td>
<td>30</td>
<td>This value determines the API name for your</td>
<td>Optional</td>
</tr>
</tbody>
</table>
Here’s a sample file that contains the header values at the beginning and lists a context segment to be imported. For importing several context segments, add more entries in the same format.

```
ApplicationId|EFFCode|ContextCode|SegmentCode|Name|ColumnName|ValueSetCode|DisplayType|Prompt|ShortPrompt|EnabledFlag|RequiredFlag|ReadOnlyFlag|Description|UomClass
0|FLEX_SN_EFF1|ColdSourceTargetContext|Cold_TEST_SEG1|cold seg 1|ATTRIBUTE_CHAR3|EFF_BASIC_FMT_CHR|TEXT_BOX|Prompt|Short Prompt|Y|Y|N|TEST desc aug 14_3PM
```

Related Topics
- Overview of Files for Import and Export
- Upload Files to WebCenter Content Server

FAQs for Manage Extensible Flexfields

Why did the extensible flexfield context not appear at run time?
If a deployed extensible flexfield context doesn’t appear in the user interface, verify that the context is associated with one of the category's pages defined for the extensible flexfield.

Manage Key Flexfields

Overview of Key Flexfields

*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 supplier #450 sold by division #10 (office supplies). Behind the scenes, the application uses a unique number, 13452, for this part, but the user always sees the 10-PEN-BLA-450 part number.

The following aspects are important to understanding key flexfields:

- Architecture
- Segments and segment labels
- *Structures*
- *Segment* and *structure instances*
- Combinations
- Dynamic combination creation
- Security
Key flexfields aren’t optional. You must configure key flexfields to ensure that your applications operate correctly. You configure and maintain key flexfield definitions with the Manage Key Flexfields task. To get a list of predefined key flexfields, use the Manage Key Flexfields task in the Setup and Maintenance work area. For information about specific key flexfields, see the help for the product where the associated business component is implemented.

Architecture
Flexfield metadata is stored in the flexfield metadata tables. When you configure a key flexfield, you define metadata about the key flexfield covering aspects such as:

- Segments are in a structure
- Structures in the flexfield
- **Value sets** in each segment

Based on the flexfield metadata, actual part numbers are captured at run time as a combination of segment values and stored in a combinations table. A combinations table contains all the segment columns for a flexfield, a unique ID column, and a structure instance number column. The structure instance number column differentiates multiple arrangements of the segment columns. For example, a part number containing multiple segments can be represented by a key flexfield. A part number key flexfield has a corresponding combinations table. In that table, the flexfield stores a list of the complete codes, with each segment of the code in a column, with the corresponding unique ID and structure instance number for the code. When users define a new part number or maintain existing part numbers in the parts catalog, they directly maintain rows in the combinations table.

The foreign key table contains a different business entity than the combinations table. For example, the business entity in the foreign key table is order lines or invoice lines that contain foreign key references to parts for ordering. Any number of foreign key tables can reference a particular entity represented by a key flexfield.

Segments and Segment Labels
A key flexfield contains segments and a segment label identifies a particular segment within a key flexfield. Segment labels are defined and made available by the product development. A segment contains the following details:

- A prompt
- A short prompt
- Display width
- The sequential position of the segment within the key flexfield structure
- The range type
- Column name of the attribute being stored by the segment
- A default value set
- A label for the segment

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, the requirement is to identify which segment in the accounting flexfield contains balancing information and which segment contains natural account information. A segment label determines which segment you are using for natural account information. When you define your accounting flexfield, you must specify which segment labels 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 helps a user searching for segments, such as the Cost Center label for all segments across key flexfields that store a value for the cost center.
Structures

A key flexfield structure definition includes the number of segments and their order.

In some applications, different users like 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 users based on a data condition in your application data, such as the value of another field entered by the 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 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. To store a particular segment, such as Cost Center, in two different structures, you must define the segment separately in each structure. 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 values or value sets used in validating 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 is registered with a tree structure, you can specify a tree code for a segment instance.

Combinations

A combination is a complete code, or combination of segment values that makes up the code, that uniquely identifies an object.

For example, each part number is a single combination, such as PAD-YEL-11x14 or 01-COM-876-7BG-LTN. In these combinations, the hyphen is the segment separator. If you have ten parts, define ten combinations. A valid combination is an existing or new combination that can be used because it's currently active and doesn't violate cross-validation or security rules. A combination has different segments depending on the flexfield structure being used for that combination. Any combination is associated with only one particular flexfield structure.

Many applications refer to a key flexfield combination by using the name of the entity or the key flexfield itself. For example, Assets uses the asset key flexfield and refers to one of its combinations as an asset key or asset key flexfield. In another example, Oracle Fusion General Ledger refers 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 (an account combination ID) for that code. Then, other tables in the application have a column that stores just the unique ID for the code. For example, you may have a part number code, such as PAD-YEL-11x14. The Parts combinations table stores that code along with its ID, 57494. If your application lets you take orders for parts, you might then have an Orders table that stores orders for parts. That Orders table would contain a single column that contains the part ID, 57494, instead of several columns for the complete code PAD-YEL-11x14. Typically, one combinations page maintains the key flexfield, where the key flexfield is the representation of an entity in your application. Maintain individual combinations, such as part numbers in the combinations page.
Dynamic Combination Creation

Dynamic combination creation is the insertion of a new valid combination into a combinations table from a page other than the combinations page. The following table lists the levels at which dynamic combination creation may be enabled.

<table>
<thead>
<tr>
<th>Level Of Dynamic Combination Creation</th>
<th>Controlled By:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexfield</td>
<td>Application development</td>
</tr>
<tr>
<td>Each usage or reference to the key flexfield</td>
<td>Application development</td>
</tr>
<tr>
<td>Structure instance</td>
<td>Administrators and implementation consultants</td>
</tr>
<tr>
<td>Other</td>
<td>Administrators and implementation consultants</td>
</tr>
</tbody>
</table>

If your key flexfield or certain usages or references of the key flexfield don’t permit dynamic combination creation, you may control whether dynamic combination creation is enabled for each structure instance. If enabled, a user can enter a new combination of segment values using the flexfield window from a foreign key page. For example, when entering a transaction, a GL user can enter a new expense account combination for an account that doesn’t yet exist. Your application creates the new account by inserting the new combination into the combinations table behind the scenes. Assuming that the new combination satisfies any existing cross-validation rules, the flexfield inserts the new combination into the combinations table, even though the combinations table isn’t the underlying table for the foreign key page.

**Related Topics**
- [Update Existing Setup Data](#)

**Considerations for Planning Key Flexfields**

Your first step in planning your key flexfields is to determine which key flexfields your application requires. Your plan should include:

- The purpose of the key flexfield
- The number and length of its available segment columns
- Whether your key flexfield permits more than one structure
- Whether more than one structure must be defined
- The number, order and length of your segments for each structure

**Before You Begin**

Once you have identified a flexfield, plan its configuration in advance. Compile a list of the UI pages and other artifacts in your deployment that are affected by the configuration. Verify that you are provisioned with the roles required to view and configure the flexfield. Use the Highlight Flexfields command in the Administration menu to view the run time page where the flexfield appears. Plan how you deploy the flexfield for test and production users and review the tools and tasks available for managing flexfields.

If you plan to use value sets, create them before configuring the key flexfield. You cannot create value sets for key flexfields at the time that you add and configure key flexfield segments.
Access to Flexfield-Related Tasks
To configure flexfields and value sets, you must have access to the tasks for managing flexfields. Contact your security administrator for details. For information about product-specific flexfield tasks, such as Manage Fixed Assets Key Flexfields, refer to the product-specific documentation.

Restrictions
If you plan to use value sets, create them before configuring the flexfield. Plan your key flexfield configuration to scale to your enterprise needs. For example, if you expect to disable old cost centers and enable new ones frequently, plan a larger maximum size for your cost center value set so that you can have more available values. A 3-character value set with one thousand available values provides more room for changes than a 2-character value set with 100 available values.

Note the code name of the flexfield you intend to configure so that you find it easily in the tasks for managing key flexfields. In some cases you can configure how the flexfield appears on the page. See product-specific documentation to determine any restrictions on using product-specific key flexfields.

Reporting
To report on your data by certain criteria or sub-entities, such as account number or project or region, consider making that sub-entity a distinct segment, rather than combining it with another sub-entity. You can categorize and report on smaller discrete units of information.

Considerations for Managing Key Flexfields
Consider the plans for a key flexfield, security, and resulting run time pages when configuring key flexfields.

Planning
Plan structures carefully and enable them for future needs. Don’t change the number, order, and maximum length of segments once you have acquired flexfield data.

Structure Delimiters
A delimiter separates the segments when they appear to 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.

Identify the delimiter value of your key flexfield carefully so that it doesn’t conflict with the flexfield data. For example, if your data frequently contains periods, such as in monetary or numeric values, don’t use a period as your segment separator. Any character you expect to appear frequently in your segment values or descriptions isn’t a good choice for the delimiter. If you change the configuration of a key flexfield, such as the delimiter, the change affects the previously stored key flexfields with that structure.

Security
Oracle Fusion data security enforces value set security.

Within key flexfields, value set security applies to the selection of the individual segment values in the segment list of values. When selecting a key flexfield segment value from the combinations table, data security permits display of only the combinations whose segment values you have access to. Applications development controls whether or not value set security rules propagate to the foreign key table. By default they do.
Run Time Pages

Application development determines the user interface (UI) pages used to render flexfields. The types of key flexfield UI pages are as follows:

- Combinations pages where the underlying entity objects use the combinations table itself
- Foreign key pages where the underlying entity objects contain a foreign key reference to the combinations table
- Partial usage pages where some or all of the key flexfield's segment columns are in a product table

The same key flexfield can be used in different ways on different pages.

A page with a foreign key reference has a base table or view that contains a foreign key reference to a combinations table with the actual flexfield segment columns. This lets you manipulate rows containing account combination IDs (account combination).

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, only a part of the configuration is likely to be visible. This enables the key flexfield to act more like a descriptive flexfield.

An account combination maintenance page or combinations page presents the combinations table. This enables directly creating and maintaining account combinations. The combinations table contains all key flexfield segment columns and a unique ID column.

A typical application has only one combinations page. An application might not have a combinations page if it doesn't support maintenance by administrators.

A page containing a search region enables 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, 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, 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 options for selecting combinations.

For more information about key flexfield pages, see the Oracle Fusion Applications Developer's Guide.

Key Flexfield Structures

A key flexfield structure arranges the segments of a key so that you can reuse a single key flexfield in multiple combinations of the same segments or a subset of those segments. Multiple instances of a single structure can accommodate differences in the value sets assigned to the structure’s segments.

The structure determines the following aspects of a key flexfield:

- The segments to include
- The order of the segments
- Segment labels on the included segments
- Properties for each segment applied to the instances of the segments in an instance of the structure
Managing Key Flexfield Structures
All the segments defined for a key flexfield are available to be included in a key flexfield structure.
You can define as many segments as there are defined segment columns in your key flexfield combinations table. Ensure that you add segments in the order that your key requires. Once deployed, the order cannot be changed.
Enable segments to indicate that they are in use. A flexfield doesn't display disabled segments in run time. 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
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

The differences among structure instances include whether dynamic combination creation is permitted. Likewise, at the structure instance level, differences among segment instances are based on the following:
- Value set
- Default type and default value
- Tree code
- Whether the segment is any of the following:
  - Required
  - Displayed
  - Enabled for business intelligence
  - Optional or required as a query criterion

For example, you can use one group of value sets for the US and another for France.
The following 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 share all the properties defined for that structure. However, the structure instances may vary if the properties are defined at the structure instance or segment instance level. For example, the value set assigned to the segment instances.

**Query Required Segment Instances**

You can designate a key flexfield segment instance as a query for making it a selectively required attribute. A user can use it as a key flexfield combination search. On the Manage Key Flexfields UI page, if you indicate that a segment instance requires indexing, add the column representing the segment to the database index. Commonly, a database administrator (DBA) adds columns to the database index.
Following deployment, the combination picker of the key flexfield displays the query required attributes as selectively required. A user must specify at least one of the query required attributes in the search criteria. This prevents unnecessary searches that could cause performance issues.

For example, you mark the cost center and account attributes as query required and ensure that the corresponding columns in the database are indexed. A user can search for combinations by entering cost center or account or both as search criteria. No search is performed if a user doesn't enter at least one query required attribute as search criteria.

**Tip:** Index the Structure Instance Number column on your combinations table to improve run time performance.

### Dynamic Combinations

If a **key flexfield** supports dynamic combination creation, you can select to enable this feature by selecting **Dynamic Combination Creation Allowed**. As a result, users enter values at run time that produce new account combinations for the flexfield. If **Dynamic Combination Creation Allowed** isn't enabled, new valid combinations can only be entered using the combinations table for the flexfield.

### Trees

You may define a tree code for the value set assigned to the segment instance. When 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. The tree structure may be fixed across all segments in the flexfield, or may vary across segments.
- A tree code for that **tree structure** exists.
- The tree code includes tree versions containing the values of the value set assigned to the segment instance.
- You assign the required tree code directly to the segment instance.

If these conditions are satisfied, you can assign the same or different tree codes to the different segment instances that use the same value set.

### Cross-Validation Rules

You can control the creation of new key flexfield code combinations by defining cross-validation rules. A cross-validation rule defines validation across segments and enforces whether a value of a particular segment can be combined with specific values of other segments to form a new combination.

The following table compares segment validation to cross-segment validation:

<table>
<thead>
<tr>
<th>Type of validation</th>
<th>Type of control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment validation</td>
<td>Controls the values you can enter for a particular segment</td>
</tr>
<tr>
<td>Cross-segment validation</td>
<td>Controls the combinations of values that administrators and end users can create for key flexfields</td>
</tr>
</tbody>
</table>

**Note:** You can use cross-validation rules for any key flexfield that has cross-validation enabled. See the documentation for your key flexfield to determine if it supports cross validation.
Cross-validation rules prevent the creation of combinations with values that can't coexist in the same combination. For example, your company requires that all revenue accounts must have a specific department. Therefore, account combinations that have revenue account values, such as all values between 4000 and 5999, must have a corresponding department value other than 000, which indicates no department is specified. You can define cross-validation rules that disallow creation of combinations with incompatible segments, such as 4100-000 or 5000-000.

Alternatively, suppose your accounting key flexfield has an Organization segment with two possible values, 01 and 02. You also have a Natural Account segment with many possible values, but company policy requires that Organization 01 uses the natural account values 001 to 499 and Organization 02 uses the natural account values 500 to 999. You can create cross-validation rules to ensure that users cannot create a general ledger account with combinations of values such as 02-342 or 01-750.

The following aspects are important to understanding cross-validation rules:

- Rule Definitions
- Enforcement
- Timing

Rule Definitions
The following table contains definitions used in cross-validation rules:

<table>
<thead>
<tr>
<th>Rule Definition</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Uniquely identifies cross-validation rules in a deployment.</td>
</tr>
<tr>
<td>Description</td>
<td>Helps administrators identify the purpose of the rule.</td>
</tr>
<tr>
<td>Error message</td>
<td>Explains why the attempted combination violates the rule.</td>
</tr>
<tr>
<td>Start Date, End Date</td>
<td>Indicates the period of time when the rule is in effect.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Determines whether the rule is enforced.</td>
</tr>
<tr>
<td>Condition filter</td>
<td>Determines the conditions in which an enabled cross-validation rule should be evaluated.</td>
</tr>
<tr>
<td>Validation filter</td>
<td>Determines the validation that the rule enforces when that condition is met.</td>
</tr>
</tbody>
</table>

When the event specified in the condition filter is applicable, the validation filter condition must be satisfied before the combination can be created. If the event specified in the condition filter isn't applicable, then the combination is considered to pass the rule and the rule won't be evaluated even if it is enabled.

**Note:** If you don't specify any statement in the condition filter, then the condition is always true and the rule is always evaluated.
Enforcement
Cross-validation prevents creation of invalid combinations by administrators using maintenance pages and end users using dynamic insertion in foreign key pages.

Enabled rules are enforced when there is an attempt to create a new combination of segment values. Disabled rules are ignored. Deleting the rule has the same effect, but you can re-enable a disabled rule.

Timing
When users attempt to create a new combination, the key flexfield evaluates any cross-validation rules that are enabled and in effect.

| Note: Cross-validation rules have no effect on combinations that already exist. The flexfield treats any existing invalid combinations that pre-date the rule as valid. |

If you want to prevent users from using previously existing combinations that are no longer valid according to your cross-validation rules, manually disable those combinations using the combinations page for that key flexfield.

When defining a cross-validation rule, specify a start and end date to limit the time when the rule is in effect. The rule is valid for the time including the From and To dates.

Considerations for Cross-Validation Rules
To validate the key flexfield combinations of segment values across segments, optimize your cross-validation rules to improve the experience of administrators and users.

Consider the following when defining cross-validation rules:

- Filters
- Rule Complexity
- Maintenance

Filters
A cross-validation rule includes a condition filter and a validation filter. The rule is evaluated using the following logical order: If the condition filter is satisfied, then apply the validation filter.

The condition filter describes the event when the rule is evaluated. If the event specified in the condition filter isn't applicable, then the rule isn't evaluated, even if enabled. When the event specified in the condition filter is applicable, the validation filter condition must be satisfied before the combination can be created.

For example, your organization has determined that a certain company value called Operations can't use a specific cost center called Marketing. You can define a cross-validation rule to validate your combinations.

1. The rule evaluates the company condition filter.
2. When company is equal to Operations, the rule evaluates the cost center validation filter.
3. When cost center is equal to Marketing, the rule prevents a combination from being created.
4. The error message you defined for the rule displays to inform the user that the attempted combination violates the rule.

Such a rule doesn't affect the creation of combinations with Marketing cost center and company values other than Operations.

Rule Complexity
For optimal performance and ease of understanding, define several simple validation rules instead of using one complex rule. Simple validation rules let you provide a more specific error message and are easier to maintain over time.
Avoid rules that control validation across more than two segments, where possible. While you can define cross-validation rules that span two or more segments, it becomes difficult to interpret cross-validation error messages and rectify invalid key flexfield combinations.

Maintenance
To maintain consistent validation, review existing key flexfields when you update your cross-validation rules. Regardless of your current validation rules, you can use an existing key flexfield combination if it's enabled. Therefore, to ensure accurate validation, you must review your existing combinations and disable any combinations that don't match the criteria of your new rules.

Tip: To keep this type of key flexfield maintenance to a minimum, decide upon your cross-validation rules when you first set up your key flexfield structure. Define cross-validation rules before creating combinations and before combinations are used in transactions.

To prevent users from using existing combinations that are no longer valid according to your cross-validation rules, disable them using the combinations page.

Overview of Cross-Validation Rules in General Ledger
You can use cross-validation rules to determine the valid account combinations that can be dynamically created as users enter transactions or journal entries. Once enabled, a cross-validation rule determines whether a selected value for a particular segment of an account combination can be combined with specific values in other segments to form a new account combination.

For example, your organization has determined that the company Operations can't use the cost center Marketing. You can define a cross-validation rule such that, if the company is Operations, then validate that the cost center isn't Marketing. New account combinations have to satisfy all of the cross-validation rules enabled for the chart of accounts before they can be created.

Entry and Maintenance
You can create cross-validation rules in the Setup and Maintenance work area using the following tasks:

- Offering: Financials
- Functional Area: General Ledger
- Task: Create Cross Validation Rules in Spreadsheet
- Offering: Financials
- Functional Area: Financial Reporting Structures
- Task: Manage Cross-Validations Rules

Use the Create Cross Validation Rules in Spreadsheet task to quickly enter large volumes of rules during implementation. Use the Manage Cross-Validation Rules task to add a one-off rule or to edit existing rules. To edit the error messages for cross-validation rules, use the following task in the Setup and Maintenance work area:

- Offering: Financials
- Functional Area: Financial Reporting Structures
- Task: Manage Messages for General Ledger

Tip: When you export or import cross-validation rules to a new instance using an export or import project in the Functional Setup Manager, you must add the Manage Messages for General Ledger task before the Manage Cross-Validation Rules task. You must export or import the messages before exporting or importing the cross-validation rules.
Existing Account Combinations

If account combinations already exist that violate newly enabled cross-validation rules, those account combinations continue to be valid. Before disabling existing account combinations that violate your rules and that you no longer use, move the balances in those accounts to the correct accounts. Then disable the account combinations manually to prevent further posting. Best practice is to define and enable cross-validation rules before: account combinations are created, transactions or journal entries are imported or entered, balances are loaded.

Related Topics
- Create Cross-Validation Rules in a Spreadsheet
- How Cross-Validation Rule Violations Are Managed
- Update Existing Setup Data

Considerations for Enabling Key Flexfield Segments for Business Intelligence

A key flexfield 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's available for use in Oracle Business Intelligence.

The following aspects are important in understanding BI-enabled key flexfield segments.

- Flattening business components to use BI-enabled segments in Oracle BI
- Equalizing segments to prevent duplication and complexity in the flattened component
- Mapping attributes of flattened business components to logical objects in Oracle BI
- Managing the labels that map segments to logical objects in Oracle BI

After you deploy a business intelligence-enabled flexfield, use the Import Oracle Fusion Data Extensions for Transactional Business Intelligence process to import the flexfield changes into the Oracle Business Intelligence repository. Users can make use of the newly-generated attributes in business intelligence applications. For additional information about logical objects and import, refer to the Oracle Transactional Business Intelligence Administrator's Guide.

Flattening

When you deploy a business intelligence-enabled key flexfield, the deployment process generates an additional set of flattened business components for use in business intelligence. The flattened business components include attributes for business intelligence-enabled segment instances only.

If you assigned a label to a segment, the flattened components include a single attribute representing all segment instances with that label. If you didn't assign a label, the flattened components include a discrete attribute for each BI-enabled segment instance in each structure.

Mapping to Logical Objects in Business Intelligence

You can simplify reporting by representing similar segments as a single logical object in Business Intelligence. If you assign a label to segments that serve the same purpose in different structures, you can consolidate the segments into a single attribute. This prevents duplication and the extra workload and complexity that result from the flattening process. For example, an organization may have more than one definition of its key accounting flexfield to support different requirements for accounting reporting. A US Accounting Flexfield structure may have a segment called Subaccount to track project expenditures. The same type of information may be tracked in a UK accounting flexfield structure with a segment called Project. Equalize these two segments to create a single list of values for reporting.

Non-labeled segments aren't equalized across context values, so the flattened components include a separate attribute for each segment for each structure. It may not be possible to equalize similarly labeled segments if they have incompatible data types or value set types.
Assign a label to a segment to map the corresponding attribute in the flattened components to a logical object in Oracle Business Intelligence. Using labels to map segments to BI logical objects minimizes the steps for importing the flexfield into Oracle Business Intelligence. 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 labels for assignment, as needed. Specify a code, name, and description to identify each label. In the BI Object Name field, enter the name of the logical object in Oracle Business Intelligence to which the segment label should map during import. Specifying the BI logical object minimizes the steps for importing the flexfield into Oracle Business Intelligence and helps to equalize context-sensitive segments across structures.

If no labels are assigned to a BI-enabled segment, or the BI Object Name on the assigned label doesn’t exist in business intelligence, you must manually map the segment to the required 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.

Importing to Oracle Business Intelligence Repository
After you deploy a business intelligence-enabled flexfield, import the flexfield changes into the Oracle Business Intelligence repository to make use of the newly flattened business components in business intelligence. Then propagate the flexfield object changes. When you import the metadata into the Oracle Business Intelligence repository, you must do so as the FUSION_APPS_BI_APPID user.

To import flexfield changes into the Oracle Business Intelligence repository in Oracle Cloud implementations, run the Import Oracle Fusion Data Extensions for Transactional Business Intelligence process. For additional information about import, refer to the Oracle Transactional Business Intelligence Administrator’s Guide.

**Note:** When you import a flexfield into the Oracle Business Intelligence repository, you see both `<name>` and `<name>_c` attributes for each segment, along with some other optional attributes. The `<name>` attribute contains the value. The `<name>_c` attribute contains the code of the value set that the value comes from, and is used for linking to the value dimension. You must import both attributes.

Example of a Key Flexfields
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 helps the user select 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 account combination (EXPENSE_LINES.EXPENSE_ACCOUNT = ACCOUNT.COMBINATION).

**Account combinations Table for Entering Accounts and Employees**
The account combinations table supports entering account information, such as for expense accounts.
The following figure shows the origin in the account combinations table of the account specified by the user. The account combination ID record stores the information of the key flexfield segments used to assemble the expense account based on the key flexfield configuration.

The combinations page, which is the maintenance page for the key flexfield, is for managing rows in the combinations table. In this example, managing the combinations means adding or editing account numbers that adhere to the key flexfield metadata rules.

The following figure shows the account combination details for the example expense account reflected in the flexfield configuration and the account combinations table.
If dynamic combination creation isn't enabled, then when entering an expense line, the user can only select an account that already exists in the ACCOUNTS (combinations) table. If they require an account that doesn't exist, they must consult with the appropriate application administrator who can add the account to the combinations table.

If dynamic combination creation is enabled, then when entering an expense line, the user can either select a preexisting account, or type in a new account that is created dynamically on the fly in the ACCOUNTS (combinations) table. Once the new combination is created, the same user can refer to it on the expense line.

When managing employee information, the user specifies the cost center that the employee belongs to. The cost center field corresponds to a single, labeled segment of the Account Key Flexfield and has metadata defined such as the allowable value set for that segment.

In the following figure, instead of specifying a cost center ID reference to an account, only the cost center segment is used and the value is stored directly on the employee table.
Define Attachments

Attachments

You can use attachments to provide supplementary information to specific business objects. Attachments can be URLs, desktop files, text, or repository folders. For a business object you may view, create, delete, or edit attachments, depending on your role and granted privileges. For more information on 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 can’t interact with the repository unless the repository mode is enabled. When enabled, users can share attachments among objects, update attachments, and perform other tasks. Access to the attachment files is controlled by a digital signing mechanism.

Security

Data security applicable to a specific business object also applies to its attachments. For example, if a user has no access to a specific expense report, then that user can’t access its attachments. You can also use attachment categories to control access and actions on attachments, based on roles associated with that category. For more information on securing attachments, see the Oracle Fusion Applications Developer’s Guide.

Attachment Entities

An attachment entity is usually a database entity, for example a table or view, that represents a business object with which attachments can be associated. Each attachment UI must be defined with a corresponding attachment entity.
Attachment entities are used only in the context of attachments and exist separately from the database entities that they're based on.

In the Setup and Maintenance work area, use the **Manage Attachment Entities** task to edit and create attachment entities. You can either use the predefined attachment entities with attachment UIs or create entities, for example when developing your own UIs.

The entity name should match the name of the table or view that represents the business object used for attachment. The name is also used in the repository folder that's automatically created to store attachments for the entity.

The *data security policies* associated with the *database resource* defined for the attachment entity apply to attachments for that entity. However, the security setting must be enabled for that entity. 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.

**Related Topics**
- Modules in Application Taxonomy
- How Database Resources and Data Security Policies Work Together

## How Attachment Entities and Attachment Categories Work Together

The association between attachment entities and categories determines the use of categories for an entity. For example, categories associated with the expense report attachment entity are available in the attachment UIs for expense reports. You can configure the associations when managing either entities or categories. Between the Manage Attachment Entities and Manage Attachment Categories pages, any change in association on one page automatically reflects on the other page. You can open either page by starting in the Setup and Maintenance work area and searching for the attachment tasks.

### Managing Entities

On the Manage Attachment Entities page, you determine which attachment categories are relevant to a particular entity. Each entity must have at least one category. For a particular expense report page with attachments functionality, you can specify which category to use for the attachment. Accordingly, the data security defined for each category is applied to the attachments on that page if security is enabled.

### Managing Categories

If you create an attachment category and must assign it to multiple attachment entities, use the Manage Attachment Categories page. The association is the same as that on the Manage Attachment Entities page.

## Attachments Troubleshooting

Attachments UIs are very user-friendly and easy to work with. You may encounter issues in certain cases such as you modify the attachments, for example create additional attachment categories, or implement data security on them.

### Issue: Can't View, Add, Update, or Delete Attachments

You may encounter the following issues when trying to view attachments or perform actions such as adding attachments.

- You can no longer see specific attachments that were earlier visible.
• You can no longer update or delete attachments.
• You get an error stating that you don't have permission to add attachments.

Resolution
Use the Manage Attachment Entities page to ensure that attachment categories are associated to the relevant attachment entity. 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, verify that the Enable Security check box is selected in the Manage Attachment Entities page for that entity. Also, make sure that users have a role that has the necessary privileges. The following table lists the privileges required to view, add, update, or delete attachments with a specific attachment category.

<table>
<thead>
<tr>
<th>Action</th>
<th>Privilege</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>Read Application Attachment (FND_READ_APPLICATION_ATTACHMENT_DATA)</td>
</tr>
<tr>
<td>Add or Update</td>
<td>Update Application Attachment (FND_UPDATE_APPLICATION_ATTACHMENT_DATA)</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete Application Attachment (FND_DELETE_APPLICATION_ATTACHMENT_DATA)</td>
</tr>
</tbody>
</table>

For example, if users have the Read Application Attachment privilege for all categories associated with the expense report attachment entity, except the Receipts attachment category, then they can view all expense report attachments except those created with the Receipts category. Likewise, if users don't have the Update Application Attachment privilege for any attachment categories tied to the expense report attachment entity, then they can't create any attachments for the expense reports.

For more information on attachment category data security, see the Oracle Fusion Applications Developer's Guide.

Certain attachments UI have predefined restrictions for users on categories. 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
You can view existing attachments but the attachments no longer have an attachment category associated with them.

Resolution
When the attachment was added, at least one category existed for the corresponding attachment entity. Since then, the entity was edited so that it no longer has any assigned categories, so the user can't 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 may need to check with your system administrator or help desk to determine the exact entity used on the page with the expenses attachments or any additional categories to assign.
Certain attachments UI have predefined restrictions for users on categories. 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?
You must use an attachment category to classify and secure an attachment. While adding attachments, you can view the available attachment categories and add the attachment to one of them. For example, attachments for an expense report can be categorized as receipts, scanned invoice images, and so on.

In the Setup and Maintenance work area, use the Manage Attachment Categories task. You can associate roles with categories to restrict user access and actions for an attachment entity. You can also create and manage categories for your own purpose, involving specific attachments with specific security requirements. For more information on attachment category data security, see the Oracle Fusion Applications Developer's Guide.

Related Topics
- Modules in Application Taxonomy
11 External Integration

Web Services

Overview of Web Services

Use web services to integrate web-based applications into your Oracle Applications Cloud. Web services expose business objects and processes to other applications using open standards-based technologies.

Web services support development environments and clients that comply with these open standards:

- Extensible Markup Language (XML)
- Simple Object Access Protocol (SOAP)
- Business Process Execution Language (BPEL)
- Web Services Description Language (WSDL)
- XML Schema Definitions (XSD)

Oracle Applications Cloud has these two types of web services:
- Application Development Framework (ADF) services
- Composite services

Let's look at the web service types in detail.

<table>
<thead>
<tr>
<th>Web Service Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADF services</td>
<td>ADF services usually represent business objects, such as employees or purchase orders. You can use these services to expose standard operations, such as create, update, and delete. However, for locally-persisted objects, ADF services aren't limited to these operations.</td>
</tr>
<tr>
<td></td>
<td>Here are a few examples of ADF services and what they're used for:</td>
</tr>
<tr>
<td></td>
<td>• Worker.changeHireDate - Use to update the hire date of the worker business object.</td>
</tr>
<tr>
<td></td>
<td>• ProjectTask.createTask - Use to adds a task to the project task business object.</td>
</tr>
<tr>
<td>Composite services</td>
<td>Composite services usually represent end-to-end business process flows that act on business events produced by the ADF services. You can use these services to manage multiple object-based services, rules services, and human workflows. Here are a few examples of composite services and what they're used for:</td>
</tr>
<tr>
<td></td>
<td>• ProjectStatusChangeApproval.process - Use to accept the change in project status.</td>
</tr>
<tr>
<td></td>
<td>• ScheduleOrchestrationOrderFulfillmentLineService.scheduleOrders - Use to schedule resources used to fulfill an order.</td>
</tr>
</tbody>
</table>

For more information about web services, see the SOAP Web Services guide for your cloud services.
Developer Connect

Overview of Developer Connect
The Developer Connect portal provides information about the web services deployed to your Oracle Applications Cloud instance. You can use this information to integrate with or extend Oracle Applications Cloud and develop solutions to suit your business requirements.

To use the Developer Connect portal, your job role must have the `FND_INTEGRATION_SPECIALIST_JOB`, `FND_APPLICATION_DEVELOPER_JOB`, or `ZCA_CUSTOMER_RELATIONSHIP_MANAGEMENT_APPLICATION_ADMINISTRATOR_JOB` privilege. Alternately, you can grant the `ATK_WEB_SERVICE_INFO_ACCESS_PRIV` entitlement to your custom role, and then grant the custom role to users who require access. For more information, see the security guide for the appropriate cloud service at Oracle Help Center (https://docs.oracle.com).

To open the Developer Connect portal, from the Navigator menu, select **Tools - Developer Connect**. This portal displays dynamic information of the web services, and the modifications done to web services to integrate with Oracle Applications Cloud. You can synchronize the Developer Connect portal with your cloud instance to retrieve the latest web service information such as service attributes, operations, business objects, security policies, and WSDL files.

Use the Developer Connect portal to:

- View the summary of the web service information such as the business object that the service defines, life cycle status, and security policy.
- Discover the operations available for the selected web service, and the request and response payloads for each operation.
- View the hierarchy of the service data objects and know information such as the data type and whether it's a required field for user-defined attributes.
- Review the sample payload XMLs for the operations of the web service. You can add or edit sample payloads, and also delete user-defined sample payloads.

Web Service Life Cycle
A web service goes through three phases in a life cycle:

- **Active**: A service is active when it's delivered the first time, until it's deprecated.
- **Deprecated**: A service is deprecated when it's superseded by a newer version, or if there is a planned obsolescence in a future release. A service in this state is still supported and becomes obsolete in a later release.
- **Obsolete**: A service is obsolete when it's no longer shipped or supported.

| Note: You must use the active version of the service. If you were previously using a service that has been deprecated, then migrate to the new, active version. |

Message Patterns
All operations exposed on a business object service have both synchronous and asynchronous message patterns defined. For conciseness, the service documentation includes the definition for the synchronous message pattern only. Both synchronous and asynchronous operations have the same functional behavior, and request and response payloads. Custom object services don't have corresponding asynchronous operations.
Naming Conventions and Examples
The naming convention for the asynchronous operation is:

- Operation name: Synchronous operation name appended with Async
- Callback name: Synchronous operation name appended with AsyncResponse

Using Help Topic Abstract Service as an example, if the name of the synchronous operation is getEntityList, the asynchronous operation name and callback name would be getEntityListAsync and getEntityListAsyncResponse.

Manage Web Service Sample Payloads
This example demonstrates how to add and edit a sample payload XML for a web service operation. It also describes how to delete a user-defined sample payload XML.

Adding a Sample Payload XML
Follow these steps to add a sample payload XML for the getEntityList operation of Help Topic Abstract Service:

1. From the Navigator menu, select Tools - Developer Connect.
2. On the Web Service overview page, search for Help Topic Abstract Service, and click the service display name.
3. On the Summary page, click the Sample Payloads tab, and then click Add Sample Payload.
4. Select getEntityList operation from the list, and enter a brief description.
5. Enter the payload XML, and click OK.

Editing a Sample Payload XML
The Developer Connect portal displays predefined and user-defined payloads of the web services. You can edit only the user-defined payloads. Follow these steps to edit a user-defined sample payload of the getEntityList operation of Help Topic Abstract Service:

2. On the Summary page of the web service, click the Sample Payloads tab and select getEntityList operation.
3. In the Edit Sample Payload dialog box, edit the payload XML and click OK.

Deleting a Sample Payload XML
You can delete only user-defined payloads, and not predefined payloads. Follow these steps to delete a user-defined sample payload of the getEntityList operation of Help Topic Abstract Service:

2. On the Summary page of the web service, click the Sample Payloads tab and select getEntityList operation.
3. Click the delete icon for the selected operation and click OK.

Import Value Set Values to Oracle Applications Cloud
This example demonstrates how to use the information in the Developer Connect portal to import value set values to Oracle Applications Cloud.

Suppose a fictional partner application wants to extend it with certain functionality available in another application. Instead of creating the required business objects and attributes in this application, you can import them using a web service. Before you proceed with the import process, enable the access permissions for the web service and review the information in the Developer Connect portal:

- Synchronize the web services information
- Review the web service details
Synchronizing Web Services Information

You must synchronize the Developer Connect portal with your Oracle Applications Cloud instance to get the latest web service information. Follow these steps:

1. From the Navigator menu, select Tools > Developer Connect.
2. Click Synchronize.

**Note:** The Last Refreshed date indicates when the Developer Connect portal was last synchronized with your cloud instance.

Searching And Reviewing Web Service Information

On the Web Services overview page, you can enter Manage File Import and Export in the Find field. You can also use Advanced Search option to find web service names that contain import, and select the relevant service from the search results.

To review the information:

1. On the Web Services overview page, select Manage File Import and Export Service. The Summary panel shows information such as the display name, business object that the service defines, life cycle status, QName, security policy used, and a brief overview of the service.
2. Click the Operations tab to view the operations supported by the web service.
3. Click the uploadFiletoUCM operation and review the result parameter in the response payload. This parameter holds the file ID in the WebCenter Content repository from which the value set values are imported.
4. Click the valueSetValuesDataLoader operation and review the fileIdAtRepository parameter in the request payload. This parameter holds the file ID in the WebCenter Content repository.
5. Click WSDL File link of the service to download the web service information.

Updating Request Payload

To add a sample payload to import the value set values:

1. Click the Sample Payloads tab and then click Add Sample Payload.
2. Select uploadFiletoUCM from the operation name list.
3. Enter a brief description of the payload in the description text box.
4. Add the payload to get the file ID from the WebCenter Content repository:

   ```xml
   <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
     xmlns:ns1="http://xmlns.oracle.com/oracle/apps/fnd/applcore/webservices/types/"
     xmlns:ns2="http://xmlns.oracle.com/oracle/apps/fnd/applcore/webservices/"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
     <soap:Body>
       <ns1:uploadFiletoUCM>
         <ns1:document xsi:type="ns2:DocumentDetails">
           <ns2:fileName>VS123.txt</ns2:fileName>
           <ns2:contentType>plain/text</ns2:contentType>
           <ns2:content>...</ns2:content>
         </ns1:document>
       </ns1:uploadFiletoUCM>
     </soap:Body>
   </soap:Envelope>
   ```

---

**Note:** The file content in the above example is a placeholder and should be replaced with the actual content. The `VS123.txt` file should be replaced with the file you want to import and export through the web service.
Cross-Origin Resource Sharing

CORS

Cross-Origin Resource Sharing (CORS) is a mechanism that allows cross-domain communication and lets a browser securely access resources from a different domain. By default, browser-based programming languages, such as JavaScript, can access resources only from the same domain. But with CORS, you can overcome this limitation and manage resources across domains.

Here are the CORS headers you can configure to make that possible.

<table>
<thead>
<tr>
<th>CORS Header</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access-Control-Allow-Origin</td>
<td>Contains a comma-separated list of trusted origins, or domains, that a client application can get resources from.</td>
</tr>
<tr>
<td>Access-Control-Max-Age</td>
<td>Specifies how long to store the results of a request in the preflight result cache.</td>
</tr>
<tr>
<td>Access-Control-Allow-Methods</td>
<td>Contains a comma-separated list of HTTP methods allowed in a request.</td>
</tr>
<tr>
<td>Access-Control-Allow-Headers</td>
<td>Contains a comma-separated list of HTTP headers allowed in a request.</td>
</tr>
<tr>
<td>Access-Control-Allow-Credentials</td>
<td>Specifies whether a client application can send user credentials with a request.</td>
</tr>
</tbody>
</table>
Example
A client application retrieves resource X from server A, which is what the application runs on. The client application then makes an HTTP request to get resource Y from server B. For this cross-server request to work, you must configure the `Access-Control-Allow-Origin` header in server B. Otherwise, the request fails and we end up with an error message.

Related Topics
- Set Profile Option Values

Configure CORS Headers
To enable CORS in Oracle Applications Cloud, configure CORS headers so that client applications in one domain can use HTTP requests to get resources from another domain. Set values for profile options that correspond to the CORS headers. In the Setup and Maintenance work area, use the `Manage Applications Core Administrator Profile Values` task in the Application Extensions functional area.

CORS Profile Options
This table lists the profile options you can set for CORS headers.

<table>
<thead>
<tr>
<th>CORS Header</th>
<th>Profile Option Name (Profile Option Code)</th>
<th>Profile Option Values</th>
</tr>
</thead>
</table>
| `Access-Control-Allow-Origin`  | `Allowed Domains (ORACLE. ADF. VIEW. ALLOWEDORIGINS)` | Here’s what you can enter to indicate which origins are allowed:
  - URL of the specific origin, for example, http://www.exampledomain.com
  - `*` to allow access to resources from all origins
  - Nothing (leave it blank) to prevent access to resources from any origin
  
  **Note:** You must set a value for this header to enable CORS. |
| `Access-Control-Max-Age`       | `CORS: Access-Control-Max-Age (CORS_ACCESS_CONTROL_MAX_AGE)` | Default value for caching preflight request is 3600 seconds. |
CORS Header | Profile Option Name (Profile Option Code) | Profile Option Values
--- | --- | ---
Access-Control-Allow-Credentials | CORS: Access-Control-Allow-Credentials (CORS_ACCESS_CONTROL_ALLOW_CREDENTIALS) | Select True or False to allow or prevent sending user credentials with the request. The default is False.

**Related Topics**
- Set Profile Option Values

---

**View Details About Predefined Scheduled Processes**

To use web services to run predefined scheduled processes, you require details about the processes. View job definitions that the processes are based on, for example to get information about parameters. You might also need to find security requirements for running the scheduled process.

**Job Definitions**

A job definition contains the metadata that determines how a scheduled process works and what options are available during submission.

To view job definitions:

1. Go to the Setup and Maintenance work area.
2. From the Application Extensions functional area, open any of these tasks as appropriate:
   - Manage Enterprise Scheduler Job Definitions and Job Sets for Financial, Supply Chain Management, and Related Applications
   - Manage Enterprise Scheduler Job Definitions and Job Sets for Human Capital Management and Related Applications
   - Manage Enterprise Scheduler Job Definitions and Job Sets for Customer Relationship Management and Related Applications
3. In the Manage Job Definitions tab, select your job definition and click Edit.
   
   ![Note: Predefined job definitions are marked with an asterisk.]

4. Cancel after you get the information you need.

**Security**

Privileges provide the access required to run specific scheduled processes. Privileges are granted to duty roles, which are granted to job roles. To see which job roles inherit the required privileges, use the Security Console or the security reference manuals for the appropriate product family.
Files for Import and Export

Overview of Files for Import and Export

You can import data into or export data out of the applications. A repository stores the content and the import and export processes handle the data movement into and out of the repository. 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:

- Using the File Import and Export page
- Interacting with content management
- Uploading to facilitate import
- Downloading to facilitate export
- Determining the file size

The File Import and Export Page

Use the File Import and Export page to upload content to or download content from the document repository of Oracle WebCenter Content. Search criteria on the page are limited to the minimum metadata of content management records needed for file import and export. To open the page, from the Navigator in the global header, select Tools > File Import and Export.

Contact the WebCenter Content Administrator for the following additional requirements:

- Information or assistance regarding general access to content management (including all metadata)
- Creating and managing accounts
- Programmatically uploading and downloading content

Interacting with Content Management

Each user with access to the File Import and Export page is assigned to one or more accounts in Oracle WebCenter Content. Accounts organize and secure access to the content items.

Uploading to Facilitate Import

Uploading a file creates a record in Oracle WebCenter Content. When you upload a file, you must also specify an account to which you upload the file. The account you specify determines which import process picks up that file to import it. You can upload any compatible file format, such as MIME, which the content repository can parse. However, the uploaded format must conform to the requirements of the import process. For example, the comma-separated values (CSV) file for the Load Interface File for Import process.

Downloading to Facilitate Export

Records in the search results table of the File Import and Export page provide download links to the files.

File Size

Upload and download don't apply the following by default:

- Data compression
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).

Guidelines for File Import and Export

Interaction between the File Import and Export page and Oracle WebCenter Content requires securing content in an account. You can use the predefined accounts that are available in Oracle WebCenter Content.

Areas of file import and export involve the following:

- Defining security
- Searching records
- Accessing content in a new account
- Naming the account
- Deleting files

Defining Security

You require the File Import and Export Management duty role for accessing the File Import and Export page. This duty role is included in the predefined role hierarchy for integration specialist roles and product family administrator roles. Files in Oracle WebCenter Content are associated with an account so that only users having access to that account can work with those files. Account names are unique and each account is treated as discrete by access control. You can only upload and download files to and from content repositories that are linked to the accounts you can access. The underlying integrated content management handles security tasks such as virus scanning.

Searching Records

A record in Oracle WebCenter Content contains the metadata used for accessing the file. When a scheduled process is run on a file, the record for the file is assigned a process ID.

Accessing Content in a New Account

After you create a new account in Oracle WebCenter Content, restart the content server. Otherwise, when you use the File Import and Export page to access content in the new account, you may experience a delay. The policy store is being updated with the new account information, which causes the delay.

Naming the Account

If you create accounts for importing or exporting data, use the following conventions for naming the account:

- Don’t include a slash (/) at the beginning or end.
- End the name with a dollar sign ($) to avoid partial string matching.
- Use dollar sign and slash ($/) as a separator in the hierarchical structure.

For example: `fin$/journal$/import$`. The File Import and Export page transforms account names by removing the dollar sign ($) separators. For example `fin$journal$import$` appears as `fin/journal/import`. The Remote Intradoc Client (RIDC) HTTP command-line interface (CLI) transforms the account name you specify without the dollar sign ($) to one that includes the sign. For example, `fin/journal/import` becomes `fin$/journal$/import$` in WebCenter Content.
Deleting Files
You can delete one file at a time when you use the File Import and Export page. To delete multiple files simultaneously from the content repository, use the standard service page in Oracle WebCenter Content.

External Data Integration Services for Oracle Cloud

Overview of External Data Integration Services for Oracle Cloud

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.

External Data Integration Services for Oracle Cloud include the following components:

- Templates to structure, format, and generate the data file according to the requirements of the target application tables.
- File-based load process to load the data files into the interface tables.
- Application-specific data import processes to transfer data from interface tables to the application tables in your Oracle Fusion Applications.
The following flow diagram outlines the steps involved in loading data from external sources.

For further information, see Using External Data Integration Services for Oracle ERP Cloud (2102800.1) on My Oracle Support at https://support.oracle.com.

**Related Topics**
- Using External Data Integration Services for Oracle ERP Cloud

**File Import Templates**

The File Based Data Import guides in the Oracle Help Center (http://docs.oracle.com) include integration templates to help you prepare external data for loading and importing. Each template includes table-specific instructions, guidelines, formatted spreadsheets, and best practices for preparing the data file for upload. Use the templates to ensure that your data conforms to the structure and format of the target application tables.

Preparing external data using templates involve the following tasks:
- Downloading templates
- Preparing data using the XLS template
Downloading Templates

To download the templates:

1. Open the File Based Data Import guide for your cloud service.
2. Locate the import process.
3. View the list of files.
   - Control files describe the logical flow of the data load process.
   - XLSM templates include the worksheets and macros for structuring, formatting, and generating your data file.

   Note: You can use XML templates to import data into Oracle Data Integrator.

4. Click the template link in the File Links table to download the file. For example, click JournalImportTemplate.xlsm in the Journal Import topic.

Preparing Data Using the XLS Template

To prepare your data in a spreadsheet format:

1. Open the XLS template. The first worksheet in each file provides instructions for using the template.

   Note: If you don’t follow the instructions, you get data load errors and data import failures.

2. Save the file.
3. Click the Generate CSV File button.

   The macro generates a comma-separated values (CSV) file and compresses the file into a ZIP file. You must transfer the ZIP file to the content management server.

Opening the XML Template

To prepare your data in Oracle Data Integrator, download the XML templates using the following steps:

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:
   - Integration projects
   - Content management document transfer utility
7. Execute the package. The package generates the CSV file and compresses it into a ZIP file.

Use Excel Integration Templates to Generate Data Files

The File Based Data Import guides in the Oracle Help Center (http://docs.oracle.com) include integration templates to help you prepare external data for loading and importing. Each template includes table-specific instructions, guidelines,
formatted spreadsheets, and best practices for preparing the data file for upload. Use the templates to ensure that your data conforms to the structure and format of the target application tables.

**Template Structure**

The integration templates include the following characteristics:

- 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.
- Columns are formatted, where applicable, to match the target field data type to eliminate data entry errors.

The worksheet columns appear in the order that the control file processes the data file.

For more information on the template structure, see the Instructions and CSV Generation worksheet in the template.

**Template Requirements**

To minimize the risks of an unsuccessful data load, ensure the following:

- Unused columns can be hidden, but not reordered or deleted.

  **Caution:** Deleting or reordering columns causes the load process to fail and results in an unsuccessful data load.

- External data must conform to the data type accepted by the control file and process for the associated database column.
- Date column values must appear in the YYYY/MM/DD format.
- Amount column values can’t have 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.
- For columns that require internal ID values, refer to the bubble text for additional guidance about finding these values.

After you finish preparing the data in the sheet, click the **Generate CSV File** button to generate a ZIP file containing one or more CSV files.

**Use XML Templates to Generate Data Files for Integration**

The File Based Data Import guides in the Oracle Help Center (https://docs.oracle.com) include XML integration templates that you use with Oracle Data Integrator to generate import files from your external data. Oracle Data Integrator provides a solution for integrating complex data from a variety of sources into your Oracle Fusion applications.

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: For Oracle Cloud implementations, you must upload the ZIP file to the content management repository in Oracle Cloud. For non-Cloud implementations, you can streamline the data integration process by installing the content management document transfer utility, which uses Oracle Data Integrator to transfer the ZIP file.

Use XML Integration Templates to Generate Data Files

Use XML templates in Oracle Data Integrator to prepare your external data for the load and import process.

The File Based Data Import guides in the Oracle Help Center (https://docs.oracle.com) include three types of XML templates that you import as target models in your Oracle Data Integrator repository:

- 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:

- Use the family-level XML file to support assets in the family, for example, Oracle Fusion Financials or Human Capital Management.
- Import the family-level XML file into your Oracle Data Integrator repository prior to importing the other XML files.
- Import one family-level XML file as a model folder for each family of products.
- Import each family-level XML file as a high-level model folder.
- Import the family-level XML file one time; it supports all subsumed product-level model folders.
- 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:

- Use the product-level XML file to support assets in the product line, for example, Fixed Assets, General Ledger, or Payables.
- Import one product-level XML file as a model folder for each line of products.
- Import the product-level XML file as a model folder into your Oracle Data Integrator repository.
- Import the family-level XML file before you import product XML files.
- Import each product-level XML file as a mid-level model folder within the appropriate family-level model folder.
- Import the product-level XML file one time; it supports all subsumed product models.
- Select Synonym Mode Insert Update as the import type.

Product XML Files

A product XML file represents a specific interface table asset.

Consider the following points when you use product XML files:

- Import one product XML file as a model for each interface table or set of tables, for example, Mass Additions.
• Import the product XML file as a model into your Oracle Data Integrator repository after you import the product-level XML file.
• Import each product XML file as a model within the appropriate product-level model folder.
• Import each product XML file one time. The model is based on File technology.
• Select Synonym Mode Insert Update as the import type.
• After you import the product model, connect the model to the correct logical schema.

How You Create Integration Projects That Generate Data Files for Import

When you use Oracle Data Integrator to generate the import data files from external data sources, you must configure an integration project. Integration projects are collections of ODI components that provide the procedural details of an integration from a source to a target. The source is your external data and the target is the import data file that you load and import into your Oracle Fusion Applications.

To create your integration project, you configure the following components:

• Knowledge modules
• Integration interfaces

Knowledge Modules

Knowledge modules contain the information that Oracle Data Integrator requires to perform a specific set of tasks against a specific technology or set of technologies. For example, check knowledge modules ensure that constraints on the sources and targets are not violated, and integration knowledge modules load data to the target tables.

Consider the following points about knowledge modules:

• Knowledge modules that you import into your integration project depend on the source and target technologies, as well as other integration-specific rules and processes.
• Multiple types of knowledge modules exist in ODI.
• Use the SQL File to Append module to create the import data file.

Integration Interfaces

Integration interfaces contain the sets of rules that define the loading of data from one or more sources to the target.

Consider the following points about integration interfaces:

• The source is the data store from your external data model.
• The target is the interface table data store, which is the CSV file from your interface table model.
• After you set up the source and target data stores, map the target fields to the source fields, or map source field values to target fields or constants.

Transfer Data Files to Oracle WebCenter Content Using Manual Flow

After you generate the ZIP file that contains the CSV data import file, transfer the ZIP file to the content repository.
Use any of the following methods to transfer file:

- File Import and Export page in Oracle Fusion Applications: Manual flow
- Oracle Fusion ERP Integration web service: Automated flow

Aspects of transferring data files to content management involve the following:

- Target accounts
- Accessing transferred content

Predefined Target UCM Accounts
You can transfer data files to predefined accounts in the Universal Content Management server that correspond to the interface table or assets.

To find the UCM account:

1. Open the File Based Data Import guide for your cloud service.
2. Locate your respective import process. For example, Journal Import.
3. View the UCM account in the Details section.

For more information, see the following guides in the Oracle Help Center (https://docs.oracle.com):

- SOAP Web Services guide for your cloud services
- File Based Data Import guide for your cloud services

Document Transfer Utility
The WebCenter Content Document Transfer Utility for Oracle Fusion Applications is a feature-set Java library that provides programmatic access to the content repository. Use the utility to import and export documents, such as import files that contain external data that you want to load into interface and application tables.

The library includes:

- Oracle WebCenter Content client command line tool
- Oracle Data Integrator upload and download tools
- Oracle WebCenter Content remote intradoc client (RIDC)
- Oracle HTTPClient
- Oracle Fusion Applications branding and defaults

Options for the WebCenter Content Document Transfer Utility for Oracle Fusion Applications fall into these categories:

- DownloadTool program options
- UploadTool program options
- Debugging and silent invocation options

DownloadTool Program Options
This table describes the download tool program options:
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>Protocol-specific connection URL of content server</td>
</tr>
<tr>
<td>user name</td>
<td>User name to leverage</td>
</tr>
<tr>
<td>password</td>
<td>Password, supplied in command line</td>
</tr>
<tr>
<td>passwordFile</td>
<td>Password, supplied in text file on the first line of the file</td>
</tr>
<tr>
<td>dID</td>
<td>ID of document revision to download</td>
</tr>
<tr>
<td>dID</td>
<td>dID is unique across repository</td>
</tr>
<tr>
<td>dID</td>
<td>dID changes with each revision</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Alternatively, specify the dDocName and RevisionSelectionMethod to identify the dID to leverage.</td>
</tr>
<tr>
<td>dDocName</td>
<td>Content name</td>
</tr>
<tr>
<td>dDocName</td>
<td>Multiple revisions of a document can share the same dDocName value, otherwise it is unique.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You should also provide RevisionSelectionMethod value.</td>
</tr>
<tr>
<td>RevisionSelectionMethod</td>
<td>Revision to download</td>
</tr>
<tr>
<td>RevisionSelectionMethod</td>
<td>Valid values: Latest, LatestReleased</td>
</tr>
<tr>
<td>RevisionSelectionMethod</td>
<td>Default value: Latest</td>
</tr>
<tr>
<td>outputFile</td>
<td>Path and name of local file to write</td>
</tr>
</tbody>
</table>

Here you see a sample download invocation command:

```java
java -classpath "oracle.ucm.fa_client_11.1.1.1.jar" oracle.ucm.client.DownloadTool
url=http://ucmserver.com:16200/cs/idcplg username=weblogic password=welcome
id=21537 outputFile="/tmp/output.doc"
```

Here you see sample output:

```
Oracle WebCenter Content Document Transfer Utility
Oracle Fusion Applications
Copyright (c) 2013, Oracle. All rights reserved.
Performing download (GET_FILE) ...
Download successful.
```
UploadTool Program Options

This table describes the upload tool program options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>Protocol-specific connection URL of content server</td>
</tr>
<tr>
<td>username</td>
<td>User name to leverage</td>
</tr>
<tr>
<td>password</td>
<td>Password, supplied in command-line</td>
</tr>
<tr>
<td>passwordFile</td>
<td>Password, supplied in text file on the first line of the file</td>
</tr>
<tr>
<td>primaryFile</td>
<td>Fully-qualified path of local primary file to upload</td>
</tr>
<tr>
<td>dDocAccount</td>
<td>Destination account</td>
</tr>
<tr>
<td>dDocTitle</td>
<td>Document title</td>
</tr>
<tr>
<td>checkout</td>
<td>If uploading a document revision, check out the document from the repository before uploading the revision. Valid values: true, false. Default value: false.</td>
</tr>
<tr>
<td>ignoreCheckoutErrorNeg22</td>
<td>Ignore error -22 (user has already checked-out the document) when checking-out the document. Valid values: true, false. Default value: true.</td>
</tr>
</tbody>
</table>

Here you see a sample upload invocation command:

```
java -classpath "oracle.ucm.fa_client_11.1.1.jar" oracle.ucm.client.UploadTool
url=http://ucmserver.com:16200/cs/idcplg username=weblogic password=we1com3i
primaryFile="/tmp/resume.doc" dDocTitle="Resume of MSMITH" -dDocAccount=/acme/sales
```

Here you see sample output:

```
Performing upload (CHECKIN_UNIVERSAL) ...
Upload successful.
[dID=21537 | dDocName=UCMFA021487]
```
Debugging and Silent Invocation Options

This table describes the usable options which are common to all tools.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>verbose</td>
<td>Verbose output</td>
</tr>
<tr>
<td></td>
<td>Log filled with Request/ Response DataBinders</td>
</tr>
<tr>
<td>quiet</td>
<td>Minimal output</td>
</tr>
<tr>
<td>version</td>
<td>Print tool revision or version</td>
</tr>
<tr>
<td>log_file_name</td>
<td>Send program output to specified log file instead of the System.out log file</td>
</tr>
<tr>
<td>log_file_append</td>
<td>Append log to existing log file rather than overwrite it</td>
</tr>
<tr>
<td></td>
<td>Valid values: true, false</td>
</tr>
<tr>
<td></td>
<td>Default value: false</td>
</tr>
<tr>
<td>socketTimeout</td>
<td>Override time out of socket</td>
</tr>
<tr>
<td></td>
<td>Specify override time in seconds</td>
</tr>
</tbody>
</table>

You can use the tools to test the connection. Provide only the url, user name, and password as you see in this sample test:

```
java -classpath "oracle.ucm.fa_client_11.1.1.jar" oracle.ucm.client.DownloadTool
url=http://ucmserver.com:16200/cs/idcplg username=weblogic password=we1com3i
```

Here you see the sample output:

```
Oracle WebCenter Content Document Transfer Utility
Oracle Fusion Applications
Copyright (c) 2013, Oracle. All rights reserved.
Performing connection test (PING_SERVER) ...
Connection test successful.
```

Load Interface File for Import Process

Use to load external setup or transaction data from a data file in the content repository to interface tables. The process prepares the data for import into application tables.

You run this process from the Scheduled Processes page. You can run it on a recurring basis.

Before running this process, you must:

1. Prepare your data file.
2. Transfer the data file to the content repository.

Parameters

Import Process
Select the target import process.

Data file
Enter the relative path and the file name of the *.zip data file in the content repository.

Import Data into Application Tables
The final destination for your external data is the application data tables of your Oracle Fusion application. Importing data into application tables involves 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 application 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 using the following steps:

1. From the Navigator, click **Tools**.
2. Click **Scheduled Processes**.
3. Click the **Schedule New Process** button.
4. Search and select the Load Interface File for Import job.
5. On the Process Details page:
   a. Select the target import process.
   b. Enter the data file name.
6. Submit the process.

If the process is successful, the status is SUCCEEDED and the process populates the interface tables. If the process isn’t successful, the status is ERROR.

**Note:** The data file remains in the content repository after the process ends.

Finding and Submitting the Import Process
Run the appropriate import process to import the data into the interface tables of your Oracle Fusion application.

To import your data into the application tables:

1. From the Navigator, click **Tools**.
2. Click **Scheduled Processes**.
3. Click the **Schedule New Process** button.
4. Search and select the import process for the target application tables.
5. On the Process Details page, select the process that corresponds to the data that you’re importing. For example, Journal Import.

If you prepared your data using the spreadsheet template, select the process shown in the Overview section of the spreadsheet.

6. Submit the process.

If the process is successful, the status is SUCCEEDED. The data in the interface tables is validated and the successful records are imported into the Oracle Fusion application tables. If the process isn’t successful, the status is ERROR.

**Note:** For more 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 Load File to Interface child process ends as an error or warning. All rows that were loaded by the process are deleted and the entire batch of records is rejected.

### Correcting Data Upload Errors

To correct errors:

1. Review the error logs.
2. Change any structural or formatting anomalies in the data.
3. Generate the ZIP file containing the CSV files using the template.
4. Upload the file to the UCM server and resubmit the Load Interface File for Import process.
5. Repeat these steps until the process successfully loads all the data.

### Correcting Import Process Errors

If the import process fails with errors:

1. Review the errors in the import log.
2. Correct the error records using the ADFdi correction spreadsheets.

### Deep Links

You can use deep links to open pages without navigating through the menu structure. Deep links come in handy if you have corporate internal portals and you want to enable direct navigation from the portals into the Oracle Fusion Applications. For example, you can enable direct navigation to the My Team page. Use the Deep Links work area to view a complete list of the available deep links.

You can simply copy a URL from the Deep Links page and paste it in your external portals as is. You don’t even need to know the URL format.

### Login and Security

Users with the View Administration Link (FND_VIEW_ADMIN_LINK_PRIV) privilege can view the Deep Link menu item in the Navigator. If you click a deep link URL from an external application and have not yet signed in, you’re automatically redirected to the Sign In page.
12 Moving Common Reference Objects

Overview of Moving Common Reference Objects

The common reference objects 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's created for an implementation project. While moving the application content, for example, moving from test to the production phase of an implementation, attend 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 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, you must note all the dependencies before moving the objects so that there are no broken references among them.

Related Topics

- Overview of Setup Data Export and Import
- Setup Data Export and Import Using an Offering or a Functional Area

Business Objects for Moving Common Reference Objects

Common reference objects in Oracle Fusion Functional Setup Manager are used to move application setup content from one environment to another. For example, from a test environment to a production environment.

Choice of Parameters

The following table lists the business objects, the movement details, and the effect of the setup task parameter on the scope of the movement.

<table>
<thead>
<tr>
<th>Note:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• You can move only the translations in the current user language.</td>
</tr>
<tr>
<td>• You can move the Oracle Social Network business objects and the changes to the Navigator using the configuration sets on the Configuration Set Migration page.</td>
</tr>
<tr>
<td>Business Object Name</td>
</tr>
<tr>
<td>---------------------------------</td>
</tr>
<tr>
<td>Application Message</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Application Taxonomy</td>
</tr>
<tr>
<td>Application Attachment Entity</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Application Attachment Category</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Application Document Sequence Category</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Application Document Sequence</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Business Object Name</td>
</tr>
<tr>
<td>-----------------------------</td>
</tr>
<tr>
<td>Application Descriptive Flexfield</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Application Extensible Flexfield</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Application Key Flexfield</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
## Moving Common Reference Objects

<table>
<thead>
<tr>
<th>Business Object Name</th>
<th>Moved Functional Item</th>
<th>Effect on the Scope of Movement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Flexfield Value Set</td>
<td>Value set setup data</td>
<td>automatically submits affected flexfields for redeployment. Only flexfields with a deployment status of Deployed or Deployed to Sandbox are eligible to be moved.</td>
</tr>
<tr>
<td>Application Reference Currency</td>
<td>Currency data</td>
<td>No parameters: All currencies are moved.</td>
</tr>
<tr>
<td>Application Reference ISO Language</td>
<td>ISO language data</td>
<td>No parameters: All ISO languages are moved.</td>
</tr>
<tr>
<td>Application Reference Industry</td>
<td>Industry data including industries in territories data</td>
<td>No parameters: All industries are moved.</td>
</tr>
<tr>
<td>Application Reference Language</td>
<td>Language data</td>
<td>No parameters: All languages are moved.</td>
</tr>
<tr>
<td>Application Reference Natural Language</td>
<td>Natural language data</td>
<td>No parameters: All natural languages are moved.</td>
</tr>
<tr>
<td>Application Reference Territory</td>
<td>Territory data</td>
<td>No parameters: All territories are moved.</td>
</tr>
<tr>
<td>Application Reference Time zone</td>
<td>Time zone data</td>
<td>No parameters: All time zones are moved.</td>
</tr>
<tr>
<td>Application Standard Lookup</td>
<td>Standard lookup types and their lookup codes</td>
<td>No parameters: All standard lookups are moved.</td>
</tr>
</tbody>
</table>

Parameter moduleType/ moduleKey
Only standard lookups belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.

Parameter valueSetCode: Only the specified value set is moved.

Importing the metadata of a value set can change the deployment status of flexfields that use the value set. Therefore, you must redeploy if there are any affected flexfields. The import process automatically submits affected flexfields for redeployment.
<table>
<thead>
<tr>
<th>Business Object Name</th>
<th>Moved Functional Item</th>
<th>Effect on the Scope of Movement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Common Lookup</td>
<td>Common lookup types and their lookup codes</td>
<td>Parameter lookupType: Only the specified common lookup is moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No parameters: All common lookups are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameter moduleType/ moduleKey Only common lookups belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameter lookupType: Only the specified common lookup is moved.</td>
</tr>
<tr>
<td>Application Set-Enabled Lookup</td>
<td>Set-enabled lookup types and their lookup codes</td>
<td>Parameter moduleType/ moduleKey Only set-enabled lookups belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameter lookupType: Only the specified set-enabled lookup is moved.</td>
</tr>
<tr>
<td>Application Profile Category</td>
<td>Profile categories</td>
<td>Parameter moduleType/ moduleKey Only categories belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>name/ applicationId Only the specified category is moved.</td>
</tr>
<tr>
<td>Application Profile Option</td>
<td>Profile options and their values</td>
<td>Parameter moduleType/ moduleKey Only profile options and their values belonging to the specified module are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameter profileOptionName: Only the specified profile option and its values are moved.</td>
</tr>
<tr>
<td>Application Profile Value</td>
<td>Profile options and their values</td>
<td>Parameter moduleType/ moduleKey Only profiles and their values belonging to the specified module are moved.</td>
</tr>
<tr>
<td>Business Object Name</td>
<td>Moved Functional Item</td>
<td>Effect on the Scope of Movement</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Application Reference Data Set</td>
<td>Reference data sets</td>
<td>No parameters: All sets are moved.</td>
</tr>
<tr>
<td>Application Reference Data Set Assignment</td>
<td>Reference data set assignments</td>
<td>Parameter determinantType: Only assignments for the specified determinant type are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameter determinantType/referenceGroupName Only assignments for the specified determinant type and reference group are moved.</td>
</tr>
<tr>
<td>Application Tree Structure</td>
<td>Tree structures and any labels assigned to the tree structure</td>
<td>No parameters: All tree structures (and their labels) are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameter moduleType/moduleKey Only trees belonging to the specified module are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameter treeStructureCode: Only trees belonging to the specified tree structure are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameter TreeStructureCode/TreeCode Only trees belonging to the specified tree structure and tree code are moved.</td>
</tr>
<tr>
<td>Application Tree</td>
<td>Tree codes and versions</td>
<td>No parameters: All trees are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameter moduleType/moduleKey Only trees belonging to the specified module are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameter treeStructureCode: Only trees belonging to the specified tree structure are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameter TreeStructureCode/TreeCode Only trees belonging to the specified tree structure and tree code are moved.</td>
</tr>
<tr>
<td>Application Tree Label</td>
<td>Tree structures and any labels assigned to the tree structure</td>
<td>No parameters: All tree structures (and their labels) are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameter moduleType/moduleKey Only tree structures (and their labels) belonging to the specified module and its descendant modules in the taxonomy hierarchy are moved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameter treeStructureCode: Only the specified tree structure (with its labels) is moved.</td>
</tr>
</tbody>
</table>
### Guidelines for Moving Related Common Reference Objects

Certain common reference objects may use other common reference objects creating dependencies among the objects. During the movement of common reference objects, ensure that these dependencies or references aren't broken or lost.

#### 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 decide 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 decide to move only value sets, or move both value sets and their lookups as part of the same package. Whatever the combination, Oracle recommends 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 following order:

1. Move created taxonomy modules before moving any objects that reference them, such as flexfields, lookups, profiles, messages, and so on.
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 have dependencies on newly created security roles, you must move the security roles before moving the security policies.

### Guidelines for Moving Common Reference Objects Using the Seed Data Framework

To move the common reference objects, you can use the Seed Data Framework. You can also use the command line interface of the Seed Data Framework 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 Seed Data Framework to import flexfield metadata, the flexfield setup metadata is imported into your database. However, you must initiate the flexfield deployment process separately after seed data import to regenerate the runtime flexfield artifacts in the target environment. Similarly, if you use Seed Data Framework 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 isn't lost during the movement, certain guidelines are prescribed. It's recommended that you perform the movement of object data exactly in the following order:

**Note:** Only the translation in the current user language is moved.

1. Move created taxonomy modules before moving any objects that reference them, such as flexfields, lookups, profiles, attachments, reference data sets, document sequences, messages, and data security.
2. Move created currencies before moving any objects that reference them, such as territories.
3. Move created territories before moving any objects that reference them, such as languages and natural languages.
4. Move created ISO languages before moving any objects that reference them, such as languages, natural languages, and industries.
5. Move created tree structures before moving any objects that reference them, such as trees or tree labels.
6. Move created profile options before moving any objects that reference them, such as profile categories or profile values.
7. Move created attachment entities before moving any objects that reference them, such as attachment categories that reference them.

8. Move created reference data sets before moving any objects that reference them, such as reference data set assignments and set-enabled lookups.

9. Move created document sequence categories before moving any objects that reference them, such as document sequences.

10. Move created tree labels before moving any objects that reference them, such as trees.

11. Move created data security objects and policies before moving any objects that reference them, such as value sets.

12. Move created value sets before moving any objects that reference them, such as flexfields.

13. Move created trees before moving any objects that reference them, such as key flexfields.
Glossary

accounting flexfield
The structure that determines the chart of accounts, including the number and order of the individual segments, as well as assigning the value sets to the segments.

action
The kind of access, such as view or edit, named in a security policy.

ADF
Application Developer Framework. A set of programming principles and rules for developing software applications.

analytics
Business intelligence objects such as analyses and dashboards that provide meaningful data to help with decision making.

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.

autosuggest
Suggestions that automatically appear for a search field, even before you finish typing your search term. You can select any of the suggestions to run your search.

balancing segment
A chart of accounts segment used to automatically balance all journal entries for each value of this segment.

business function
A business process or an activity that can be performed by people working within a business unit. Describes how a business unit is used.

business intelligence catalog
The repository where all business intelligence objects, including analyses, reports, briefing books, and agents, are stored. The catalog contains separate folders for personal, shared, and modified objects.

business object
A resource in an enterprise database, such as an invoice or purchase order.

business unit
A unit of an enterprise that performs one or many business functions that can be rolled up in a management hierarchy.
calendar event
A period that signifies an event, such as a public holiday or a training course, that impacts worker availability.

chart of accounts
The account structure your organization uses to record transactions and maintain account balances.

clause adoption
Reusing a clause from the global business unit in local business units either by adopting the clause without change or by localizing it.

clause localization
A type of clause adoption where the adopted clause is edited to suit the local business unit needs.

clause numbering level
Specifies the determinant type of the document sequence for automatic clause numbering.

condition
The part of 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 be associated with a different set of context-sensitive segments.

context-sensitive segment
A flexfield segment that may or may not appear depending upon a context. Context-sensitive segments are attributes that apply to certain entity rows based on the value of the context segment.

contract deviations
Differences between the contract terms in a contract and those in the contract terms template applied to that contract and any deviations from company policies as determined by Contract Expert feature rules.

Contract Expert
A feature of the application that permits you to create business rules in the Contract Terms Library to enforce corporate policies and standards for contracts.
**Contract Terms Library**
A repository of standard clauses, contract terms templates, and business rules built using Contract Expert.

**corporate rate type**
Rate you define to standardize rates used in conversion of one currency to another over a period of time. This rate is generally a standard market rate determined by senior financial management for use throughout the organization.

**cost center**
A unit of activity or a group of employees used to assign costs for accounting purposes.

**country holding company**
A legal entity that acts on behalf of several divisions within an enterprise, and is the legal employer in a country.

**dashboard**
A page that gives quick access to key tasks and summary information for a business process or object.

**data model**
The metadata that determines where data for a report comes from and how that data is retrieved.

**data security**
The control of access and action a user can take 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**
Expandable fields used for capturing additional descriptive information or attributes about an entity, such as a customer case. You may configure information collection and storage based on the context.

**determinant**
A value that specifies the use of a reference data set in a particular business context.

**determinant type**
An optional value that affects document sequencing in a transaction. The available determinant types are Business Unit, Ledger, Legal Entity, and Tax Registration.
determinant type
The value that affects sharing of reference data in a transaction across organizations, such as a business unit or a cost organization.

determinant value
A value specific to the selected determinant type of a document sequence. 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. It is relevant in a document sequence assignment only if the document sequence has a determinant type.

division
A business-oriented subdivision within an enterprise. Each division is organized to deliver products and services or address different markets.

document sequence
A unique number that is automatically or manually assigned to a created and saved document.

enterprise
An organization having common control over one or more legal entities.

entitlement
Grant of access to functions and data. Oracle Fusion Middleware term for privilege.

extensible flexfield
Expandable fields that you can use to capture multiple sets of information in a context or in multiple contexts. Some extensible flexfields let you group contexts into categories.

feature
Business practices or methods applicable to the functional areas that enable the fine-tuning of business functionality.

fixed rate type
Rate you set between two currencies that remains constant. For example, a rate set between the euro currency and each Economic and Monetary Union (EMU) currency during the conversion to the euro currency.

flexfield
A flexible data field that you can configure such that it contains one or more segments or stores additional information. Each segment has a value and a meaning.

flexfield segment
An extensible data field that represents an attribute and captures a value corresponding to a predefined, single extension column in the database. A segment appears globally or based on a context of other captured information.
global header
The uppermost region in the user interface that remains the same no matter which page you’re on.

global search
The search in the global header that lets you search across many business objects.

grade
A component of the employment model that defines the level of compensation for a worker.

import
In the context of data integration, the transfer of data from interface tables to application tables, where the data is available to application users.

incident
A collection of diagnostic information about a critical error, providing details about the state of the application when the issue occurred.

interface table
Database table that stores data during data transfer between applications or between databases that reside inside and outside of an Oracle Fusion application.

item master
A collection of data that describes items and their attributes recorded in a database file.

job
A generic role that's independent of any single department or location. For example, the jobs Manager and Consultant can occur in many departments.

job definition
The metadata that determines what a job does and what options are available to users when they submit the scheduled process. A job is the executable for a scheduled process.

job role
A role, such as an accounts payable manager or application implementation consultant, that usually identifies and aggregates the duties or responsibilities that make up the job.

key flexfield
Configurable flexfield comprising multiple parts or segments, each of which has a meaning either individually or in combination with other segments. Examples of key flexfields are part numbers, asset category, and accounts in the chart of accounts.
**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, you can define multiple structures for a single key flexfield.

**key flexfield structure instance**
An occurrence of a key flexfield structure that shares the same order of segments as other instances of the key flexfield structure. However, each instance uses different value sets to validate the segments.

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

**legal employer**
A legal entity that employs people.

**legal entity**
An entity identified and given rights and responsibilities by commercial law through the registration with country's appropriate authority.

**legal jurisdiction**
A physical territory, such as a group of countries, single country, state, county, parish, or city, which comes in the purview of a legal authority.

**legal reporting unit**
The lowest level component of a legal structure that requires registrations. Used to group workers for the purpose of tax and social insurance reporting or represent a part of your enterprise with a specific statutory or tax reporting obligation.

**legislative data group**
A means of partitioning payroll and related data. At least one legislative data group is required for each country where the enterprise operates. Each legislative data group is associated with one or more payroll statutory units.

**line of business**
Set of one or more highly related products which service a particular customer transaction or business need. Refers to an internal corporate business unit.

**load**
In the context of data integration, the transfer of external data from data files to the receiving **interface tables** in preparation for an import into application tables.
lookup code
An option available within a lookup type, such as the lookup code BLUE within the lookup type COLORS.

lookup type
The label for a static list that has lookup codes as its values.

mainline metadata
The primary branch of metadata that a sandbox is published to. Once published, changes made in the sandbox become available to all users.

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.

party fiscal classification
A classification used by a tax authority to categorize a party for a tax.

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.

position
A specific occurrence of one job that's fixed within a department. It's also often restricted to 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 of access to functions and data; a single, real world action on a single business object.

profile option
User preferences and system configuration options that users can configure to control application behavior at different levels of an enterprise.

profile option level
The category or layer that defines a profile option. Site, Product, and User are the predefined levels.

profile option value
The setting mapped to the level of a profile option. A profile option may have multiple values set at different levels, such as Site or User.
**Query By Example**
The icon for filtering data in a table.

**reference data**
Data in application tables that is not transactional or high-volume, which an enterprise can share across multiple organizations. For example, sales methods, transaction types, or payment terms.

**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 collection of reference data sets that correspond to logical entities, such as payment terms defined across multiple tables or views. Based on the common partitioning requirements across entities, the reference data sets are grouped to facilitate data sharing among them.

**registration**
The record of a party's identity related details with the appropriate government or legal authorities for the purpose of claiming and ensuring legal and or commercial rights and responsibilities.

**report**
An output of select data in a predefined format that's optimized for printing.

**role**
Controls access to application functions and data.

**sandbox**
A testing environment that isolates untested code changes from the mainline environment so that these changes don't affect the mainline metadata or other sandboxes.

**scheduled process**
A program that you run to process data and, in some cases, generate output as a report.

**segment**
A segment is a single field within a flexfield and maps to a single table column in your database. When configuring a flexfield, you define the appearance and meaning of individual segments.
service provider model
A business unit that provides specific business functions for another business unit.

set
Classified and grouped reference data that organizational entities share.

set enabled
A property that describes entities that an organization shares as reference data. For example, you can indicate a lookup, customer, location, or document attachment as set enabled.

spot rate type
Rate you enter to perform conversion based on this rate as of a specific date. This rate applies to the immediate delivery of a currency.

Style template
An .rtf template containing style information that's applied to report layout templates to achieve a consistent look and feel across reports.

Subtemplate
An .rtf or .xsl format that is defined once and used multiple times within a single report layout template or across multiple layout template files.

suggestion group
Category of suggestions that appear in the autosuggest for the global search.

tax registration
The registration of a party with a tax authority that confers tax rights and imposes certain tax obligations.

territory
A legally distinct region used in the country field of an address.

tree
Information or data organized into a hierarchy with one or more root nodes connected to branches of nodes. A tree must have a structure where each node corresponds to data from one or more data sources.

tree structure
A set of guidelines or a framework applied to create a tree, include data, version a tree, or access a tree.

tree version
An instance of a tree that includes life cycle elements such as start and end dates, and indicates whether the tree is active. If a tree is associated with a reference data set, all tree versions belong to one set.
**user rate type**
Rate you enter at journal entry time to convert foreign currency transactions to your ledger currency.

**value set**
A predefined set to validate the values that a user enters in the application. The set may be hierarchical.

**work area**
A set of pages containing the tasks, searches, and other content you need to accomplish a business goal.

**work relationship**
An association between a person and a legal employer, where the worker type determines whether the relationship is a nonworker, contingent worker, or employee work relationship.

**workflow**
An automated process that passes a task from one user (or group of users) to another to view or act on. The task is routed in a logical sequence to achieve an end result.