# Contents

## Preface

## 1 Introduction to Product Management  
Product Management Offering: Overview  
Product Lifecycle Management and Product Hub: Overview  
Implementation Tasks: Overview  
Setting up Innovation Management: Roadmap  
Setting up Product Development: Roadmap  
Setting up Product Hub: Roadmap  
Setting up Product Hub Portal for Supplier Users: Roadmap

## 2 Define Item Organizations  
Item Organizations: Overview  
Item Organizations: Explained  
Organization Trees: Explained
# Define Items

- Define Items: Overview .......................... 13
- Item Profile Options: Explained .......... 14
- Advanced Item Profile Options: Explained 15
- Units of Measure, Unit of Measure Classes, and Base Units of Measure: How They Fit Together 16
- Lifecycle Phases: Explained ............... 17
- Manufacturer Parts and Manufacturers: Explained 17
- Product Child Value Sets: Explained .... 18
- Attachment Categories: Explained ........ 18
- Operational Attributes Groups: Explained 18
- Item Attribute Groups and Attributes: Explained 19
- Item Classes: Explained .................. 20
- Item Class Descriptive Flexfields: Explained 21
- Default Item Class: Explained .......... 21
- Deploy Item Extensible Flexfields: Explained 21
- Item Statuses: Explained ................ 22
- Managing Item Types: Explained .......... 23
- Cross-Reference Types: Explained ........ 23
- Item Descriptive Flexfields: Explained 23
- Import Items: Explained .................. 25
- Generate the CSV File: Explained ....... 26
- Upload to the Universal Content Manager: Explained 26
- Import Data from the Item Management Interface Tables: Explained 27
- Monitor Item Imports: Explained .......... 28
- Related Item Subtypes: Explained ....... 28
- Item Revision Descriptive Flexfields: Explained 29
- Trading Partner Item Descriptive Flexfields: Explained 29
- Managing Document Classes: Explained 30
<table>
<thead>
<tr>
<th>4 Define Catalogs</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalogs: Overview</td>
<td>31</td>
</tr>
<tr>
<td>Catalog Descriptive Flexfields: Explained</td>
<td>31</td>
</tr>
<tr>
<td>Category Descriptive Flexfields: Explained</td>
<td>31</td>
</tr>
<tr>
<td>Create Catalog: Explained</td>
<td>32</td>
</tr>
<tr>
<td>Manage Catalogs: Explained</td>
<td>32</td>
</tr>
<tr>
<td>Functional Area Catalogs: Explained</td>
<td>33</td>
</tr>
<tr>
<td>Automatic Assignment Catalogs: Explained</td>
<td>34</td>
</tr>
<tr>
<td>Catalog Report Publishing: Explained</td>
<td>34</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5 Define Change Orders</th>
<th>37</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Order Setup: Overview</td>
<td>37</td>
</tr>
<tr>
<td>Change Order Reasons: Explained</td>
<td>37</td>
</tr>
<tr>
<td>Change Order Priorities: Explained</td>
<td>38</td>
</tr>
<tr>
<td>Change Order Statuses: Explained</td>
<td>38</td>
</tr>
<tr>
<td>Change Order Types: Explained</td>
<td>39</td>
</tr>
<tr>
<td>Manage Change Order Entry and Exit Rule Sets: Explained</td>
<td>42</td>
</tr>
<tr>
<td>Data Security Privilege for Viewing Change Orders: Explained</td>
<td>43</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6 Define Product Rules</th>
<th>45</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Rules: Overview</td>
<td>45</td>
</tr>
<tr>
<td>Rules and Rule Sets: Explained</td>
<td>45</td>
</tr>
<tr>
<td>Item Rule Data Types: Explained</td>
<td>46</td>
</tr>
<tr>
<td>Item Rule Syntax: Explained</td>
<td>47</td>
</tr>
<tr>
<td>Item Rule Multirow Attribute Group Functions</td>
<td>48</td>
</tr>
<tr>
<td>Item Rule Numeric Functions and Operators</td>
<td>49</td>
</tr>
<tr>
<td>Item Rule Production Value Functions</td>
<td>52</td>
</tr>
<tr>
<td>Item Rule String Functions</td>
<td>53</td>
</tr>
<tr>
<td>Item Rule Logical Functions and Operators</td>
<td>56</td>
</tr>
<tr>
<td>Item Rule Utility Functions</td>
<td>60</td>
</tr>
<tr>
<td>Using Custom Object Data in Rules: Explained</td>
<td>63</td>
</tr>
<tr>
<td>Blending Rules</td>
<td>64</td>
</tr>
</tbody>
</table>
### Define New Item Requests  
- New Item Requests: Overview  
- New Item Requests Workflow Statuses: Explained  
- New Item Request Definition Phase: Explained  
- New Item Request Approval Phase: Explained  
- Defining Entry and Exit Criteria for New Item Requests: Procedure

### Define Product Spoke Systems
- Product Spoke Systems: Overview  
- Product Spoke Systems: Explained

### Define Advanced Catalogs
- Advanced Catalogs: Overview  
- Catalog Mappings: Explained

### Define Item Mass Update Configuration
- Item Mass Update Configuration: Overview  
- Item Import Formats Configuration: Explained

### Define Item Import Batch Configuration
- Item Import Batch Configuration: Overview  
- Item Import Formats Configuration: Explained

### Define Audit History for Product Management
- Audit History for Product Management: Overview  
- Audit Trail: Explained

### Define Product Lifecycle Management
- Define Product Innovation  
- Define Product Development  
- Configure Integrations for Product Development
Preface

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

Oracle Applications Help

Use help icons to access help in the application. If you don’t see any help icons on your page, click the Show Help icon in the global header. Not all pages have help icons. You can also access Oracle Applications Help at https://fusionhelp.oracle.com.

Using Applications Help

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

You can also read Using Applications Help.

Additional Resources

- Community: Use Oracle Cloud Customer Connect to get information from experts at Oracle, the partner community, and other users.
- Guides and Videos: Go to the Oracle Help Center to find guides and videos.
- 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 at Oracle Accessibility Program.

Access to Oracle Support

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

Please give us feedback about Oracle Applications Help and guides! You can send e-mail to: oracle_fusion_applications_help_ww_grp@oracle.com.
1 Introduction to Product Management

Product Management Offering: Overview

Implementors can use the Product Management offering to configure services that support the following work areas in Oracle Fusion Applications:

- Ideas
- Concepts
- Portfolios
- Product Development
- Product Information Management

Note: Before you begin to set up Product Management, you must perform implementation tasks that are common to Oracle SCM Cloud offerings. For detailed information about enabling and implementing offerings, see the Implementing Common Features for Oracle SCM Cloud guide.

In the Setup and Maintenance work area, tasks are grouped under functional areas. Each functional area has a list of setup tasks. Complete the setup tasks for the functional areas that are applicable to the services you have subscribed to. These tasks are described in more detail in subsequent chapters. This table lists the offering and the associated functional areas.

<table>
<thead>
<tr>
<th>Cloud Service</th>
<th>Functional Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Innovation Management</td>
<td>• Product Requirements and Ideation Management&lt;br&gt;• Concept Design Management&lt;br&gt;• Product Lifecycle Portfolio Management</td>
</tr>
<tr>
<td>Product Development</td>
<td>• Product Development&lt;br&gt;• Product Management Business Intelligence Analytics (if your users require BI Reports for Product Development)</td>
</tr>
<tr>
<td>Product Hub</td>
<td>• Item Organizations&lt;br&gt;• Items&lt;br&gt;• Catalogs&lt;br&gt;• Structures&lt;br&gt;• Item Mass Update&lt;br&gt;• Advanced Catalogs&lt;br&gt;• New Item Requests&lt;br&gt;• Change Orders&lt;br&gt;• Product Rules&lt;br&gt;• Audit Trail&lt;br&gt;• Product Spoke System&lt;br&gt;• Item Batches&lt;br&gt;• Data Pool Integration</td>
</tr>
</tbody>
</table>

To set up the Innovation to Commercialization business process, complete the tasks for the functional areas under Innovation Management, Product Development and Product Hub.
Product Lifecycle Management and Product Hub: Overview

Oracle’s Product Lifecycle Management (PLM) solutions, along with Product Hub, can be configured to work together in order to deliver comprehensive Innovation to Commercialization capabilities across the entire product value chain. Product Lifecycle Management consists of the Innovation Management and Product Development cloud services.

- **Oracle Innovation Management**: Streamlines new product development and introduction processes with the following functional areas:
  - **Product Requirements and Ideation Management**: Allows employees and stakeholders to collaborate on product innovation ideas and record requirements. Product managers can integrate requirements with concepts in Concept Design Management, and ideas with proposals in Product Lifecycle Portfolio Management.
  - **Concept Design Management**: Offers a collaborative design workspace for product architects, designers and executives to generate, capture, analyze, and approve product concepts that address product strategy goals. Approved concepts can then be transferred directly to external PLM solutions for prototype planning, detailed design and product introduction.
  - **Product Lifecycle Portfolio Management**: Allows product portfolio managers to create, analyze, manage and revise product portfolios, to arrive at an optimal product mix.

- **Oracle Product Development**: Enables you to manage product data and change orders while balancing cost.

- **Oracle Product Hub**: Centralizes product data across heterogeneous systems so that it can create a blended product master record that is clean, standardized, accurate, and current. From products that you manufacture internally to finished goods that you source from suppliers, Product Hub enables you to aggregate, enrich, and share product data for various manufacturing and omni-channel commerce processes. Use robust business rules and workflows to make sure the data that you share across the enterprise is clean, complete, and valid. Rapidly commercialize products that use centralized product information for manufacturing, marketing, and sales across global manufacturing sites, sales channels, and trading partners.

Deploy each cloud service with configurations based on your required level of control and customization.

<table>
<thead>
<tr>
<th>Deployment Option</th>
<th>Deployed By</th>
<th>Level of Control and Customization</th>
<th>Speed of Adoption and Agility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Private Cloud</td>
<td>Oracle deploys and manages for you in an exclusive private cloud</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Oracle Public Cloud</td>
<td>Oracle provides a subscription-based service</td>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>

**Related Topics**

- Defining Product Development: Overview
- Defining Product Innovation: Overview
Implementation Tasks: Overview

This topic outlines the sequence of setup tasks for implementing Product Management. The sequence of tasks is split across the Cloud Service Administrator and Application Implementation Consultant roles.

<table>
<thead>
<tr>
<th>Tasks for the Service Administrator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1- Create a Primary Implementation User</strong></td>
<td>For your consultants to access and begin your implementation process, create the primary implementation user for your lead consultant.</td>
</tr>
<tr>
<td></td>
<td>Once completed, this user can create additional users for the rest of the implementation team.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> Oracle recommends that you set up your implementation users in the Test environment first. Migrate them to Production after they have been tested and validated.</td>
</tr>
<tr>
<td>To create the primary user, follow these instructions:</td>
<td></td>
</tr>
<tr>
<td>1. Sign in with your Oracle Fusion user ID and password.</td>
<td></td>
</tr>
<tr>
<td>2. Select <strong>Navigator &gt; Security Console</strong>.</td>
<td></td>
</tr>
<tr>
<td>3. Click <strong>Users</strong>.</td>
<td></td>
</tr>
<tr>
<td>4. Click <strong>Add User Account</strong>.</td>
<td></td>
</tr>
<tr>
<td>5. Provide the user attributes and click <strong>Add Role</strong>.</td>
<td></td>
</tr>
<tr>
<td>6. To provision the new user with roles, search for the <strong>Application Implementation Consultant</strong> role. Select it and click <strong>Add Role Membership</strong>.</td>
<td></td>
</tr>
<tr>
<td>7. Add the following <strong>Roles</strong>, at minimum:</td>
<td></td>
</tr>
<tr>
<td>o IT Security Manager</td>
<td></td>
</tr>
<tr>
<td>o Employee</td>
<td></td>
</tr>
<tr>
<td>Close the window.</td>
<td></td>
</tr>
<tr>
<td>Notify your primary implementation team member that their user ID has been created. Give them their initial password.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2- Create Implementation Projects</strong></td>
<td>Optionally, create <strong>Implementation Projects</strong> in the <strong>Setup and Maintenance</strong> work area.</td>
</tr>
<tr>
<td><strong>Step 3- Set up Key Implementation Users and Security Profiles</strong></td>
<td>After your environments are provisioned, you as the Service Administrator have sufficient security abilities to create three implementation users with the necessary roles.</td>
</tr>
<tr>
<td></td>
<td><strong>Examples:</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>TechAdmin:</strong> Can perform key technical duties, including functional setup and assigning security roles to users.</td>
</tr>
<tr>
<td></td>
<td>• <strong>APPL_IMPL_CONSULTANT</strong> and <strong>SCM_IMPL_CONSULTANT:</strong> Can perform key functional duties, including functional setup.</td>
</tr>
</tbody>
</table>
### Tasks for the Service Administrator

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>You may decide to replace or refine these initial users, but these users have all the access required to get you started.</td>
</tr>
</tbody>
</table>

### Tasks for Application Implementation Consultant

<table>
<thead>
<tr>
<th>Step 4- Create Data Roles and Assign Security Profiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
</tr>
<tr>
<td>You can secure data by provisioning roles that provide the necessary access rights.</td>
</tr>
</tbody>
</table>

Data roles apply explicit data security policies on job and abstract roles. Create and maintain data roles in the Authorization Policy Manager.

Assign a predefined security profile to relevant job or abstract roles using the Oracle Human Capital Management (HCM) setup task **Manage Data Role and Security Profiles**.

The following job and duty roles are shipped with the product. You can copy these to create additional roles as necessary:

**Innovation Management:**
- Product Design Engineer: Concept Development Duty
- Product Design Manager: Concept Management Duty
- Product Management VP: Portfolio Management Duty
- Product Manager: Product Proposal Management Duty
- Product Portfolio Manager: Portfolio Management Duty
- Employee: Idea Management Duty

**Product Development:**
- Product Manager: Product Development Duty
- Employee: Idea Management Duty

**Product Hub:**
- Product Manager
- Product Data Steward

### Step 5- Create End Users

To create end users, follow these instructions:

1. Sign in with your Oracle Fusion user ID and password.
2. Select **Navigator > Security Console**.
3. Click **Users**.
4. Click **Add User Account**.
5. Provide the user attributes and click **Add Role Membership**.
6. To provision the new user with roles, search for the relevant role and click **Add Role Membership**.

**Note:** Innovation Management includes additional steps. For more detailed information, refer to the Oracle SCM Cloud Implementing Innovation Management guide.
Tasks for Application Implementation Consultant | Description
---|---
**Step 6- Perform Common Application Configuration** | Common applications configuration includes setup of security, common reference objects, collaboration messaging, OTBI and custom ESS jobs, data export and import instructions, and maintenance tasks. For more detailed information on common implementation tasks for all SCM products, see Oracle Applications Cloud Implementing Common Features guide.

**Related Topics**
- User and Role Synchronization: Explained
- Creating Implementation Users: Procedure
- Creating Data Roles for Implementation Users: Procedure
- Enabling Offerings: Explained
- Configuring Offerings: Procedure

### Setting up Innovation Management: Roadmap

The Innovation Management configuration tasks are as follows:

<table>
<thead>
<tr>
<th>Task List</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define Product Requirements and Ideation Management</td>
<td>Use this task list to configure ideas, and requirements specifications.</td>
</tr>
<tr>
<td>- Manage Product Idea Classes</td>
<td></td>
</tr>
<tr>
<td>- Manage Product Idea Statuses</td>
<td></td>
</tr>
<tr>
<td>- Manage Product Requirements Classes</td>
<td></td>
</tr>
<tr>
<td>- Manage Product Requirements Statuses</td>
<td></td>
</tr>
<tr>
<td>- Manage Product Requirements and Ideation Lookups</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task List</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define Concept Design Management</td>
<td>Use this task list to configure concepts and concept components.</td>
</tr>
<tr>
<td>- Manage Product Concept Classes</td>
<td></td>
</tr>
<tr>
<td>- Manage Product Concept Component Classes</td>
<td></td>
</tr>
<tr>
<td>- Manage Product Concept Statuses</td>
<td></td>
</tr>
<tr>
<td>- Manage Product Concept Lookups</td>
<td></td>
</tr>
</tbody>
</table>
Task List | Description
--- | ---
Define Product Lifecycle Portfolio Management | Use this task list to configure proposals and portfolios.
  - Manage Proposal Statuses
  - Manage Product Portfolio Classes
  - Manage Portfolio Statuses
  - Manage Product Portfolio Planning Periods
  - Manage Product Portfolio Lookups
  - Manage Portfolio and Product Rule Sets

Related Topics
- Class Management in Oracle Innovation Management: Explained
- Innovation Management Lookups: Explained

Setting up Product Development: Roadmap

The Product Development configuration tasks are as follows:

Task List | Description
--- | ---
- Manage Item Organizations
- Manage Organization Trees | Use these tasks to define item organizations for Product Management.

- Manage Item Classes
  - Manage Item Attribute Groups and Attributes
  - Manage Lifecycle Phases | Use these tasks to define items.

- Manage Change Order Types
  - Manage Change Order and New Item Request Header Descriptive Flexfields
  - Manage Change Order Entry and Exit Rule Sets | Use these tasks to define change orders.

- Manage Structure Component Descriptive Flexfields
- Manage Product Development Lookups | Use these tasks to define structure component and lookups.

Related Topics
- Define Items: Overview
• Item Organizations: Overview
• Change Order Setup: Overview
• Product Development Lookups: Explained

Setting up Product Hub: Roadmap

In the Setup and Maintenance work area, tasks are grouped in functional areas. You can view and implement them through the Product Management offering. These tasks are described in more detail in subsequent chapters.

<table>
<thead>
<tr>
<th>Task List</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define Item Organizations for Product Management</td>
<td>You must define at least one item organization in order to take any actions with items. You can optionally set up organization trees to create a hierarchy of organizations to be used in various places throughout the application.</td>
</tr>
<tr>
<td>Define Items</td>
<td>There are several required and optional setup tasks that must be completed prior to working with items including:</td>
</tr>
<tr>
<td></td>
<td>• Item Profile Options: These are defined for you. You should review these settings to confirm they meet your business needs.</td>
</tr>
<tr>
<td></td>
<td>• Lifecycle Phases: You must create lifecycle phases and those must be assigned to the item class used to create the items or to a parent item class of the item class used to create the item.</td>
</tr>
<tr>
<td></td>
<td>• Product and Child Value Sets: Required for creating item rules.</td>
</tr>
<tr>
<td></td>
<td>• Attachment Categories: You can optionally define attachment categories.</td>
</tr>
<tr>
<td></td>
<td>• Attributes and Attribute Groups: Create attributes and assign them to attribute groups.</td>
</tr>
<tr>
<td></td>
<td>• Item Classes: You must create at least one item class before you can create items.</td>
</tr>
<tr>
<td></td>
<td>• Various Flexfield tasks: These are optional tasks used to gather additional item data.</td>
</tr>
<tr>
<td></td>
<td>• Deploy Item Flexfields: You must deploy flexfields after you create a new item class or make changes to any flexfields.</td>
</tr>
<tr>
<td></td>
<td>• Item Statuses: These are seeded for you.</td>
</tr>
<tr>
<td></td>
<td>• Item Types: 37 types have been seeded for you. You can edit or create additional types.</td>
</tr>
<tr>
<td></td>
<td>• Cross Reference Types: Optional task for defining cross references between two items.</td>
</tr>
<tr>
<td></td>
<td>• Download Import template, Upload Item Data, Load Interface File, Import Items, Monitor Item Imports: Use these tasks if you are importing you item data into Product Hub.</td>
</tr>
<tr>
<td></td>
<td>• Item Subtypes: Optional task for defining item subtypes.</td>
</tr>
<tr>
<td>Manage Functional Area Catalogs</td>
<td>The setup task in this task list is used to define functional area catalogs. Non-Product Hub customers use the Manage Functional Area Catalog task to create and manage catalogs. Product Hub customers complete these tasks in the Product Information Management work area.</td>
</tr>
<tr>
<td>Define Change Orders</td>
<td>Creating change order types is required before you can create change orders. You must also define the task configurations and manage the approval groups. You can optionally define change reasons, priorities and statuses.</td>
</tr>
<tr>
<td>Define Product Rules</td>
<td>You must create rules and rule sets if you plan on using product rules for validating or assigning data to items.</td>
</tr>
<tr>
<td>Define New Item Requests</td>
<td>Similar to defining change orders, you must set up task configurations and approval groups prior to creating new item requests.</td>
</tr>
<tr>
<td>Task List</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Define Product Spoke Systems</td>
<td>This task list is used to define spoke or source systems.</td>
</tr>
<tr>
<td>Define Advanced Catalogs</td>
<td>Product Hub customers can use this task list to define catalog mappings.</td>
</tr>
<tr>
<td>Define Item Mass Update Configuration</td>
<td>Used to create import formats and assign them to an item class.</td>
</tr>
<tr>
<td>Define Item Import Batch Configuration</td>
<td>Used to create import batch formats and assign them to an item class.</td>
</tr>
<tr>
<td>Define Audit History for Product Management</td>
<td>You can optionally define audit policies if you want to track who made what changes and when they were made.</td>
</tr>
</tbody>
</table>

Setting up Product Hub Portal for Supplier Users: Roadmap

Product Data Stewards need to carry out the following tasks before supplier users can begin managing their products in Product Hub Portal.

<table>
<thead>
<tr>
<th>Required Setup</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate the Job Role with the Supplier User</td>
<td>All supplier users need to be assigned the Supplier Product Administrator job role for accessing Product Hub Portal.</td>
</tr>
<tr>
<td>Define Spoke Systems</td>
<td>Separate spoke systems must be setup for every supplier who uploads product data.</td>
</tr>
<tr>
<td>Item Class Security</td>
<td>Item class security needs to be setup for the supplier users who upload product data. On the Edit Item Class page Security tab, the Supplier Product Administrator job role needs to be given item data privileges similar to other job roles such as the Product Data Steward.</td>
</tr>
<tr>
<td>Define a Default Catalog</td>
<td>The default catalog must be set using the Manage Advanced Item Profile Option task.</td>
</tr>
<tr>
<td>Create Catalog Category Mappings</td>
<td>Mappings between the categories of the default catalog and item class need to be setup for deriving the item class of supplier products.</td>
</tr>
<tr>
<td>Set up Item Extensible Attributes for Product Hub Portal</td>
<td>Use the Edit Item Class task in the Setup and Maintenance work area to expose item extensible attributes to suppliers. Attributes are selected on the Product Hub Portal sub-tab under the Pages and Attribute Groups tab of the Edit Item Class task.</td>
</tr>
<tr>
<td>Set up Import Maps</td>
<td>Import maps are used to allow suppliers to import their product data with generated pre-defined templates. Import Maps must be set to External in order for suppliers to access the associated templates.</td>
</tr>
</tbody>
</table>
Note: Supplier and Supplier User setup is handled in the Oracle Fusion Procurement application. See the Oracle Fusion Procurement documentation for more information.
### Define Item Organizations

#### Item Organizations: Overview

You must create item organizations before you can create items in Oracle Fusion Product Hub. These tasks are completed in the Setup and Maintenance work area.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Item Organizations</td>
<td>Item organizations are used when the organization does not have dependencies on business units or legal entities.</td>
</tr>
<tr>
<td>Manage Organization Trees</td>
<td>In Product Hub, organization hierarchies (trees) are used across some of the mass change flows, including the Assign to Organization, Assign Items to Supplier Organization, and Create Item Structure from Common flows.</td>
</tr>
</tbody>
</table>

#### Item Organizations: Explained

Item organizations are used to control the availability of attributes for items and item security. Item security is based on a combination of the item class, group or person, and the organization. The item organization structure is similar to the inventory organization structure, except the item organization structure does not have an association with a business unit or legal entity.

Product Management can be configured with two different organization structures:

- Item Organizations: An organization structure that does not have dependencies on business units or legal entities.
- Inventory Organizations: An organization structure that requires business units and legal entities. Used by the Oracle Fusion Material and Logistics and Procurement applications.

An item organization defines an item when inventory balances are not stored and inventory storage or inventory movement is not reflected in the Oracle Fusion Applications. For example, you would use an item organization in a retail scenario, if you need to know the items that are listed by and sold through each retail outlet even though inventory and transactions are recorded in another system. Item organizations are also used to control the availability of attributes for items and item security. The item organization structure is similar to the inventory organization structure, except the item organization structure is not required to have an association with a business unit or legal entity. The item organization also does not have the required inventory organization-level attributes.

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

For customers who have licensed only Oracle Fusion Product Hub, item organizations are sufficient.
To create an item organization:

1. Enter Organization Information: Item organizations are identified by a name and organization number. Each organization has a location which is defined by the location address. Locations are entered using the Manage Locations task. Optional information such legal entity and business unit can also be entered.

2. Enter the Item Master Organization: If this is the first item organization being created, it should be a master organization. To make this a master organization, enter the organization name again. If the new item organization is a child organization of an existing master organization, then enter the name of the master organization.

3. Enter the Starting Revision: Changes to the organization can be tracked through revisions and are usually represented by a letter or number or combination of the two.

Organization Trees: Explained

Organization trees are used to create a list of organizations for use in some of the mass change flows.

In some implementations, customers may have a specific set of organizations that they use in their business and may want to retain that list. You can define an organization tree to represent the list of hierarchies that you want to use. In Oracle Fusion Product Hub, organization trees are called organization hierarchies.

In Product Hub, organization hierarchies are used across some of the mass change flows, including Assign to Organization, Assign Items to Supplier Organization, and Create Item Structure from Common. In the mass change flows, if you want to select a set of items and assign them to a specific set of organizations, then you have the option of selecting each organization individually or selecting an organization hierarchy that represents the list of organizations. Item hierarchies are created in the Setup and Maintenance work area using the Manage Organization Trees task.
# 3 Define Items

## Define Items: Overview

Before you can define items in Oracle Fusion Product Hub, you must complete several tasks in the Setup and Maintenance work area.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Item Profile Options</td>
<td>Profile options manage configuration data centrally and influence the behavior of applications.</td>
</tr>
<tr>
<td>Manage Advanced Item Profile Options</td>
<td>This task is used by Product Hub. If you do not install Product Hub, you do not need to define these options.</td>
</tr>
<tr>
<td>Define Units of Measure</td>
<td>Units of Measure must be created before you can create or import items.</td>
</tr>
<tr>
<td>Manage Lifecycle Phases</td>
<td>Item Lifecycle Phases are used as an indicator of the stage for an item within the lifecycle process. Each phase represents a set of tasks and deliverables that are required before promoting an item to the next phase.</td>
</tr>
<tr>
<td>Manage Product and Child Value Sets</td>
<td>In Product Hub, value sets are primarily used to define attributes that have a specific set of values. Each value set is associated with one or more attributes in the same attribute group or in a different attribute group.</td>
</tr>
<tr>
<td>Manage Attachment Categories for Product Management</td>
<td>Used to create attachment categories and associate them with item classes.</td>
</tr>
<tr>
<td>Manage Operational Attribute Groups</td>
<td>Operational attributes determine the behavior of the item with respect to various applications outside of Product Hub, such as Oracle Fusion Purchasing or Oracle Fusion Inventory.</td>
</tr>
<tr>
<td>Manage Item Attribute Groups and Attributes</td>
<td>Used to determine how the attributes appear in the user interface, as well as how they are used in the application.</td>
</tr>
<tr>
<td>Manage Item Classes</td>
<td>Item classes are created at the root item class or under the parent item class and inherit values based on selections made when defining the item class. For Product Hub customers, the Manage Item Classes task is used to create and manage item classes, user defined attributes and data security.</td>
</tr>
<tr>
<td>Manage Item Class Descriptive Flexfields</td>
<td>Descriptive flexfields appear in the user interface as additional information and can also appear in search results tables.</td>
</tr>
<tr>
<td>Deploy Item Extensible Flexfields</td>
<td>After you associate attribute groups and pages with an item class, you must deploy flexfields to view the pages or attribute groups at runtime. The metadata that was created for the attribute group is not synchronized with the production data in Product Hub until the flexfield is deployed.</td>
</tr>
<tr>
<td>Manage Item Statuses</td>
<td>Item statuses are used to define the state an item is in and based on the state, the default values for item operational attributes.</td>
</tr>
</tbody>
</table>
## Task | Description
--- | ---
Manage Item Types | Item types are date effective and are made active or inactive by adjusting the start and end dates.
Manage Cross Reference Types | Cross-References provide the functionality to map additional information about an item in the form of a value and cross-reference type. For example, the cross-reference can map a relationship between an item and an old part number.
Manage Item Descriptive Flexfields | Used to define descriptive flexfields that are specific to items.
Download Import Template | Each template includes table-specific instructions, guidelines, formatted spreadsheets, and best practices for preparing the data file for upload.
Upload Item Data | After you have created the CSV file, the next step in the Import process will upload the CSV Zip file to the designated location within the Oracle Universal Content Management system.
Load Interface File through Scheduled Process | Once the CSV file is uploaded to UCM, you use the Load Interface File for Import scheduled process to move the data from the UCM folder to the interface tables.
Import Items | The Item Import task creates an Enterprise Scheduled Service (ESS) process that takes the data that is loaded in the interface tables and uses the import process to move the data to the production tables.
Monitor Item Imports | Use this task to monitor the ESS process status in the search results table.
Manage Related Item Subtypes | A related item is an item relationship between two existing items. How the two items are related is defined by a subtype.
Manage Item Revision Descriptive Flexfields | Use descriptive flexfields associated at Item Revision level to capture item revision information whose values may differ between revisions of the same item.
Manage Item Relationship Descriptive Flexfields | Item types are date effective and are made active or inactive by adjusting the Start Date and End Date.
Manage Trading Partner Item Descriptive Flexfields | When defining descriptive flexfields associated with trading partner items, you must use certain prefixes when naming the context segments, in order for the segments to be displayed for the respective trading partner type.
Define Item-Specific UOM Conversions | After you define units of measure, define the conversions used for items.
Item Profile Options: Explained

Profile options manage configuration data centrally and influence the behavior of applications. The profile options have a default value, which can be used for initial installations. These profile options should be evaluated to determine if additional values should be set:

- EGP_DISPLAY_IMAGES: Specify if images should be displayed in the search results table in the Manage Items task.
- EGP_ITEM_IMPORT_ITEMS_PER_THREAD: During item import, multiple threads of operation are created to process the items being imported. This profile option controls how many items are processed per each thread. It is a technical option used to optimize item import performance. The default value is 100.
- EGP_ITEM_IMPORT_NUMBER_OF_THREADS: This option works in conjunction with the EGP_ITEM_IMPORT_ITEMS_PER_THREAD option. It controls how many threads of operations are created during the item import process. The default is 4.
- EGP_UPDATEABLE_ITEM: By default, the item number can’t be changed after the item has been created. It can be updated after creation only if this option is set to Yes.

Advanced Item Profile Options: Explained

Profile options manage configuration data centrally and influence the behavior of applications. Only those customers who have licensed Oracle Fusion Product Hub can access these advanced profile options. The profile options have a default value, which can be used for initial installations. These advanced profile options should be evaluated to determine if additional values should be set. Manage these values using the Manage Advanced Item Profile Options task in the Setup and Maintenance work area.

- EGO_ASSIGN_PACK_ORG: Assign all child items in the pack to the same organization as the parent pack item.
- EGO_ASSIGN_PACK_SUP_SITE_ORG: Assign all child items in the pack to the same supplier site organization combination as the parent pack item.
- EGO_DEFAULT_STYLE_CATALOG: Enable the assignment of the related SKU items to the same catalog as the style item.
- EGO_DEFAULT_STYLE_ITEM__ORG: Enable the assignment of the related SKU items to the same organization as the style item.
- EGO_DEFAULT_STYLE_PEOPLE: Enable the assignment of people to the related SKU items as the style item.
- EGO_DEFAULT_STYLE_SUP_SITE_ORG: Enable the assignment of the related SKU items to the same supplier site organization combination as the style item.
- EGO_GATHER_STATS: Specify the threshold value above which statistics collection is enabled.
- EGP_ITEM_IMPORT_DEFAULT_CATG: Specify the catalog to be used for deriving the item class of items being imported.
- EGI_PUBLICATION_ITEMS_PER_PAYLOAD: Determines the number of items to be used per payload in the publication concurrent program.
- EGI_PUBLICATION_NUMBER_OF_PAYLOADS: Determines the number of parallel payloads to be used in the publication concurrent program.
• **EGO_VER_LINE_ACTIVATE_COUNT_PER_THREAD**: Specify the number of versions or change order lines that become effective in a single thread.

Units of Measure, Unit of Measure Classes, and Base Units of Measure: How They Fit Together

Define units of measure, unit of measure classes, and base units of measure for tracking, moving, storing, and counting items.

The figure below shows that the unit of measure class named ‘Quantity’ contains the units of measure: Box of 8, Box of 4, and Each. The unit of measure named Each is assigned as the base unit of measure.

Units of Measure Classes

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

Units of Measure

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

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

Related Topics

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

Lifecycle Phases: Explained

Item Lifecycle Phases are used as an indicator of the stage for an item within the lifecycle process. Each phase represents a set of tasks and deliverables that are required before promoting an item to the next phase.

Each item must have a lifecycle phase associated with it.

Four lifecycle phase types are predefined in the application: Design, Preproduction or Prototype, Production and Obsolete. You can use the predefined phase types to create new values for the lifecycle phases.

Companies may use different terms to describe the same item lifecycle phase. For example, the phases named Production and In Manufacturing both refer to the lifecycle phase during which an item can be used to build and ship products.

Lifecycle phases are associated with item classes. The items in an item class can be assigned to any of the lifecycle phases associated with that item class.

Note: The lifecycle phase of a child item is not automatically promoted when the parent item is promoted.

The effective date for items, in the Design lifecycle phase of a change order cannot be modified. If the item is in the 'Design' lifecycle phase, the change on the change order is effective on approval. If a future effective date is set for the affected object in a change order, a warning message appears with the following buttons:

1. Continue: To continue with the future effective date.
2. Edit: To cancel the current Save action, return to edit mode and correct the future effective date.

Note: The best practice is to make items Effective on Approval, if they are in the design lifecycle phase.

Before you create or import items, create lifecycle phases. Assign phases to the item class used to create the items. You can also assign them to a parent item class of the item class used to create the item. When an item is assigned to a lifecycle phase, that phase is visible as part of the item’s attributes. In item structures, lifecycle phases are used to control specific processes.

Manufacturer Parts and Manufacturers: Explained

Manufacturers must to be created before you may even create manufacturer parts produced by that manufacturer. Manufacturer parts (manufacturer items) can be related to any item designed or produced by your company. Any items on
a structure may be sourced with parts from the approved manufacturers list (AML). Manufacturer parts are identified with manufacturer part numbers (MPNs).

Add a descriptive flexfield to the manufacturer part by using the Manage Trading Partner Item Descriptive Flexfields task (in the Setup and Maintenance work area and the Product Management offering). After you add the flexfield, deploy it.

Related Topics

- Descriptive Flexfields: Explained
- Flexfield Deployment: Explained

Product Child Value Sets: Explained

The Manage Product Child Value Sets task uses the same page as the Manage Product Value Set task.

A child value set is used to define variants for stock-keeping units or SKUs. A SKU contains the common properties for an item. For example, a shirt can be produced with colors; white, red, yellow, and blue. The variant is used to represent the colors of the shirt.

You define child value sets as follows:

- Create a value set with validation type of independent, for example All Colors.
- Select the new value set in the Manage Product Value Sets results table, for example All Colors.
- Click Manage Values, create several values, for example Blue, Red, Green, Yellow, and Black.
- Create a value set with validation type of Subset and enter the first value set you created for the independent value set, for example: Summer Colors.
- Select the value set Summer Colors in the Manage Product Value Set result table.
- Click Manage Values and then click the Add icon. The dialog will show a list of values based on the value set named Summer colors. Select two of them. The value set Summer Colors is a child of All Colors.

The value set Summer Colors is a child of All Colors.

Attachment Categories: Explained

The basic tab of the Manage Item Class task is used to associate attachment categories to specific item classes.

The Attachment Categories region allows for the creation and management of attachment categories for the items created within the item class. To classify item attachments, associate attachment categories with item classes. Associated attachment categories are inherited down through the item class hierarchy.

Operational Attributes Groups: Explained

Operational attributes determine the behavior of the item with respect to various applications outside of Product Hub, such as Oracle Fusion Purchasing or Oracle Fusion Inventory.
You choose the control level for operational attributes on the Manage Operation Attribute task in the Setup and Maintenance work area. For each listed operational attribute group, you select the control level for each of the group’s attributes. You can control the operational attributes at the master organization level or at the organization level. You can define operational attributes as part of a new item request.

Some operational attributes for items are defined as key flexfields. Key flexfields allow a structured value for attribute to be captured. Key flexfields can capture a key, such as a part number, a job code, or an account code.

Examples of operational attributes with the attribute groups they belong to:

- Inventory = Shelf Life Days
- Order Management = Shippable
- Purchasing = Negotiation Required
- Receiving = Allow Substitute Receipts

Operational attributes are stored in the Items data table.

**Item Attribute Groups and Attributes: Explained**

Attribute groups are a logical group of attributes that are displayed in their own subregion of the user interface page at run time. Attribute groups can be either single-row or multiple-row. The selected behavior determines how the attributes appear in the user interface, as well as how they are used. Each attribute group is associated with one or more item classes.

To create an attribute group and attributes, you use **Manage Item Attribute Groups and Attributes** task (in the Setup and Maintenance work area under the Product Management offering). Create an attribute group specify its context usage as item. Choose the behavior of attribute as either single-row or multiple-row:

- **Single-row attribute group**: Contains a collection of attributes that appear as separate fields in a region named for the attribute group. For example, a single-row attribute group named Processor contains the attributes appropriate for a processor. When these attribute groups are displayed in the user interface, the attribute fields for each group are arranged compactly within a region titled with the name of the attribute group. Attributes can be multiple data types.

- **Multiple-row attribute group**: Attributes appear as columns in a table that represents the attribute group. Each row in the table is considered an attributes group. The attributes is collection of values specified by the columns in the table. The table appears in the user interface within a region titled with the attribute group name, such as MSRP Price. No other fields appear in the table. For example, a multiple-row attribute group named MSRP Price contains the attributes Country, MSRP, and Currency. Each row of the table describes an MSRP price, and is a value of the MSRP Price attribute group.

Within the attribute group, create an attribute and a value set. For example, create an attribute group named Cost and Compliance and within that add attribute named Material Cost. Using value sets, define the set of currencies applicable to item cost. After you create attribute group and attribute, perform the following in the Setup and Maintenance work area under the Product Management offering:

- Associate the attribute to the item class by using Manage Item Class task
- Deploy the attribute by using Deploy Item Extensible Flexfields task

Once saved, you cannot edit the behavior of the attribute group. You will have to discard it and begin the creation of new attribute group with the correct behavior type.
Related Topics

- Descriptive Flexfields: Explained
- Product Value Sets: Explained

Item Classes: Explained

Item classes are created at the root item class or under the parent item class and inherit values based on selections made when defining the item class.

The Manage Item Classes task, accessed through the Setup and Maintenance work area, is used to create and manage item classes, user defined attributes and data security.

Item classes can be defined in a hierarchy where the child levels indicate sub levels or types of the parent item class. All items are created within an item class. The item class hierarchy can be used to control processes for some levels of the hierarchy.

You can also use item classes for classification purposes and in some cases, item creation may not be allowed. By optionally setting the Item Creation Allowed attribute to No, item creation under an item class can be prevented. However, a child item class of that item class can be set to allow for item creation.

For example, this figure illustrates the Desktop item class is a child of the Computers item class and both are set to not allow item creation. Both the Green Desktop item class and the Gaming Desktop item class are children of the Desktop item class and both are set to allow item creation.

This enables you to prevent items from being created in the Computers and Desktops item class, but allows you to create items for the Green Desktops and Gaming Desktops item classes. Optionally, specify a date on which the item class will become inactive. You cannot specify an inactive date that is later than the inactive date of an item class parent, nor can you specify an inactive date that has already passed. Also, all children of a parent item class with an inactive date must be made inactive at the same time or earlier.

Note: Oracle Fusion Product Development does not support the versioning of item classes.

When setting up definition steps for a new item request at the item class, you can identify various item details as mandatory, at each step. Definition of entire entity can be made mandatory or just certain attributes. This ensures that the item information required for a downstream step is defined and available for use.

You can also inherit required attributes from the parent and assignee access is validated.
You can control item creation, view and update access by assigning a role on the item class to a principal or group of users. Security allows a person or a group to have privileges to an item of item class in each organization. This is inherited. Therefore, a person who has a privilege in a parent item class will automatically have the same privilege in the child item classes.

**Item Class Descriptive Flexfields: Explained**

User-defined attributes are used to configure additional attributes to support your organization’s requirements. Descriptive flexfields (DFF) appear in the user interface as additional information and can also appear in search results tables.

If you need to add only shallow and small numbers of individual data fields, consider using descriptive flexfields. For example, you may want to use a descriptive flexfield to capture different address fields (represented as context-sensitive segments) for different countries (represented as contexts). Address fields, though they may differ in number per country, are usually all at the same hierarchy level. For table layouts, if you have data that require a different context segment value per row, and that context segment value has different respective context-sensitive segments (in terms of type and number), then you must use descriptive flexfields, not extensible flexfields.

You cannot group attributes using descriptive flexfields. For example, if you wanted to define a maximum CPU speed and a minimum CPU Speed for an item, you have to specify an attribute called Maximum CPU Speed and another called Minimum CPU Speed. You couldn’t have a grouping called CPU Speed and have two child attributes called Maximum and Minimum.

With descriptive flexfields, you can define many contexts for an object but you can display only one context at a time. For example if the context value is a State, then the context segment called "Capital" would have different values depending on the value of the context. If the descriptive flexfields have only one context, the context selector can be hidden in the user interface. You can define descriptive flexfields on items, structures, catalogs, categories, new item requests, and change orders.

**Default Item Class: Explained**

For non-Product Hub customers, the Manage Default Item Class task (in the Setup and Maintenance work area under the Product Management offering) is used, since these customers cannot create additional item classes nor can they create user defined attributes such as EFFs. An exception to this rule is that Product Development customers can create additional item classes and EFFs. The Manage Default Item Class task launches an edit page for the Root Item Class.

The Manage Default Item Class task has three tabs:

- **Basic**: Item Class descriptive flexfields and attachment categories are defined on this tab.
- **Item Management**: Item number generation method is defined using this tab.
- **Lifecycle Phases**: The lifecycle phases that the items assigned to this item class will use are defined on this tab.
- **Item Templates**: The item templates that are used to create items are defined on this tab.
Deploy Item Extensible Flexfields: Explained

After you associate attribute groups and pages with an item class, you must deploy flexfields in order to view the pages or attribute groups at runtime. The metadata that was created for the attribute group is not synchronized with the production data in Product Hub until the flexfield is deployed.

To deploy flexfields, select the Deploy Item Extensible Flexfields task in Setup and Maintenance work area. All extensible flexfields for Product Hub are created within the flexfield code EGO_ITEM_EFF.

The deployment process is a CPU-intensive process, and is usually run at off-hour periods. You can choose from these deployment options:

- **Deploy Flexfield**: Online incremental deployment. The deployment process begins immediately. Only the extensible flexfield setup that changed is deployed.
- **Deploy Offline**: Allows the deployment to be scheduled. The flexfields are deployed, one at a time, in the order that you deploy them to the queue. Because all Product Hub extensible flexfields use the same flexfield code, the process deploys all of the attribute groups and attributes for all of the context usages at the same time. You cannot select individual attribute groups or item classes for deployment. You should choose to deploy offline if the flexfield changes impact 30 or more item classes.
- **Refresh and Deploy Offline**: Use this option only if the first two options result in errors. You must log out and log back in to view the extensible flexfield configuration on the item.

Item Statuses: Explained

In the Item Status table, select a status code to display the associated attribute groups and attributes as well as control information.

Item statuses are used to define the state an item is in and based on the state, the default values for item operational attributes.

Item statuses are seeded; the values are **Active** and **Inactive**. You can create, edit or delete item statuses on the Manage Item Statuses page.

Operational attribute groups and attributes corresponding to the selected item status are displayed in the **Details** section.

Whenever the status is applied to the item, the value of the attribute may change. Select the usage that corresponds to how the attribute value will change based on the item status value:

- **Defaulted** - Allows you to override the value during the import and update of an item.
- **Inherited** - Sets the values of the item status attributes when the status value changes. You cannot override the value.
- **None** - The item status attribute values will not be changed.

Any change made to an item status is not applied automatically to existing items. The change will be applied when the item status value is changed while editing an item.

Status attributes for each item status control the actions that you can perform on the item. Some of the status attributes are: Build In WIP, Customer Orders Enabled and Internal Orders Enabled.
Implementing Product Management

Chapter 3

Define Items

The **Controlled at** field is not editable and is populated from the value set on the Manage Attribute Groups page.

**Managing Item Types: Explained**

Item types are managed using the Manage Item Types task in the Setup and Maintenance work area (under the Product Management offering).

There are 32 seeded item types and you can edit them or create additional item types.

Item types are date-enabled and are made active or inactive by adjusting the Start Date and End Date.

To benefit from the use of item types, you must enable them by selecting the **Enable** check box.

**Cross-Reference Types: Explained**

Cross-references provide the functionality to map additional information about an item in the form of a value and cross-reference type. For example, the cross-reference can map between an item and an old part number, where the value is the value for the old part number and the type is Old Part Number. Cross-reference types are part of item relationships where the item relationship type is cross-reference. There are no values seeded for cross-reference types. You define the values using the **Manage Cross Reference Types** task. Cross-reference types are date-enabled and can be made active or inactive by adjusting the values of the start date and end date. To use the item relationship for cross-reference, you must enable cross-reference types by checking the **Enable** check box.

**Item Descriptive Flexfields: Explained**

You can use descriptive flexfields to capture additional information about items beyond what is provided by the predefined set of operational attributes in Oracle Fusion Product Hub.

**Item Descriptive Flexfields**

If you are not using Oracle Fusion Product and Catalog Management, then you cannot create user-defined attribute groups and attributes. However you can use descriptive flexfields associated at Item level to create fields to capture information about items. Like other descriptive flexfields, item descriptive flexfields have context segments and context-sensitive segments whose values are validated on entry by value sets. You can define the value sets to control what values users can enter in a descriptive flexfield segment. Examples of information that you might capture are size and volumetric weight.

Manage this flexfield type by using the Manage Item Descriptive Flexfields task available in the Setup and Maintenance work area.

**Item Revision Descriptive Flexfields**

Use descriptive flexfields associated at Item Revision level to capture item revision information whose values may differ between revisions of the same item.

Manage this flexfield type by using the Manage Item Revision Descriptive Flexfields task available in the Setup and Maintenance work area.
Item Relationship Descriptive Flexfields

When defining descriptive flexfields associated with item relationships, you must use certain prefixes when naming the context segments, in order for the segments to be displayed for the respective relationships.

The prefixes required for naming the context segments are listed in the following table, with their corresponding item relationship types. For example, if you define an item relationship descriptive flexfield with a context segment named `RELATED_RELATIONSHIP_ATTRIBUTES`, then the value segments of this context will be displayed for Related Item Relationships when users conduct transactions in that context. For another example, when users navigate to a UI of a particular object, such as a Competitor Item, they see the contexts whose internal name has the prefix `COMP`.

<table>
<thead>
<tr>
<th>Relationship Type</th>
<th>Prefix for Context Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Competitor Item Relationship</td>
<td>COMP</td>
</tr>
<tr>
<td>Customer Item Relationship</td>
<td>CUST</td>
</tr>
<tr>
<td>Item Cross-reference Relationship</td>
<td>XREF</td>
</tr>
<tr>
<td>GTIN Relationship</td>
<td>GTIN</td>
</tr>
<tr>
<td>Manufacturer Part Number Relationship</td>
<td>MFG</td>
</tr>
<tr>
<td>Related Item Relationship</td>
<td>RELATED</td>
</tr>
<tr>
<td>Source System Item Relationship</td>
<td>SYS</td>
</tr>
</tbody>
</table>

Manage this flexfield type by using the Manage Item Relationship Descriptive Flexfields task available in the Setup and Maintenance work area.

Trading Partner Item Descriptive Flexfields

When defining descriptive flexfields associated with trading partner items, you must use certain prefixes when naming the context segments, in order for the segments to be displayed for the respective trading partner type.

The prefixes required for naming the context segments are listed in the following table, with their corresponding trading partner types. For example, if you define a trading partner item descriptive flexfield with a context segment named `COMP_TPI_ATTRIBUTES`, then the value segments of this context will be displayed for Competitor Item when users conduct transactions in that context.

<table>
<thead>
<tr>
<th>Trading Partner Type</th>
<th>Prefix for Context Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Competitor Item</td>
<td>COMP</td>
</tr>
<tr>
<td>Customer Item</td>
<td>CUST</td>
</tr>
<tr>
<td>Manufacturer Item</td>
<td>MFG</td>
</tr>
</tbody>
</table>
Manage this flexfield type by using the Manage Trading Partner Item Descriptive Flexfields task available in the Setup and Maintenance work area.

**Related Topics**

- Descriptive Flexfields: Explained
- Managing Descriptive Flexfields: Points to Consider
- Flexfields: Overview
- Item Relationships: Explained

**Import Items: Explained**

Item Management provides the ability to create and management item data through two methods: the Product Information Management work area and a process to import Items from files located in a specific folder in Oracle Universal Content Management system.

The objects listed below are supported through both methods:

- Items
- Item revisions
- Item category assignments
- Item associations*
- Item relationships
- Item extensible flexfields *
- Item translatable extensible flexfields*
- Item revision extensible flexfields*
- Item revision translatable extensible flexfields*
- Item supplier extensible flexfields*
- Item translatable supplier extensible flexfields*
- Item style variant attribute value Sets
- Trading partner items

**Note:** You must license Oracle Fusion Product Hub to use extensible flexfields.

The following is an overview of the item import process:

1. Download import template.
2. Enter data in tabs within the ItemImportTemplate.xslm template file.
3. Generate CSV (ZIP file).
4. Upload to Oracle Universal Content Management.
5. Move the data into Item Management interface tables.
6. Import data to Item Management product tables.
Generate the CSV File: Explained

After downloading the template, enter data into the ItemImportTemplate.xlsx template file and generate the CSV file. The first tab in the file contains the instructions for using the template and generating the CSV ZIP file. The other tabs, also called Control Files, correspond to each of the item interface tables.

Keep these tips in mind when entering data into the template:

- The first row in each sheet contains column headers that represent the interface table columns. The columns are in the order that the control file expects them to be in the data file.
- Do not change the order of the columns in the Excel sheets. Changing the order of the columns will cause the load process to fail.
- You can hide columns that you do not intend to use, but you cannot delete columns. Deleting columns will cause the load process to fail.
- You must enter data that conforms to what the control file can accept and process for the associated database column.
  - Date fields must be entered in YYYY/MM/DD format.
  - Amount columns must not contain a thousands separator and must use a period (.) as the decimal separator.
  - Columns that must be whole numbers have data validation to allow only whole numbers to be entered.
- Refer to the bubble text on each column header for information on the data and data type the column requires.
- The provided source system code, PIMDH, is used for imports. The source system represent external systems to Product Hub.
- The provided item class is called Root Item Class, this item class will be used for imports.
- Items are always created in a master organization. Organizations are created during initial setup of the environment using the Setup and Maintenance task.
- Provide a numeric Batch ID. Non-Product Hub customers don’t have access to create batches, so any ID is sufficient.

Generate the CSV ZIP file

After you complete entering data in the ItemImportTemplate.xlsx template file, you next generate a .CSV file to be used for importing the data.

Click the Generate .CSV button on the Instruction and .CSV Generation tab to generate the file. This will create a .ZIP file containing a CSV file for each tab completed in the ItemImportTemplate.xlsx file.

You will be prompted to save the file to the desktop location several times. The first is to save the file, the second will ask the folder where the .ZIP file will be saved and the next set of save requests will be for each of the tabs.

Upload to the Universal Content Manager: Explained

After you have created the CSV file, the next step in the Import process will upload the CSV Zip file to the designated location within the Oracle Universal Content Management system.
UCM is an enterprise content management system that manages web content, documents, imaging management, and digital assets for the Oracle Fusion Applications.

**Note:** You must have the correct functional privileges to support the upload of the CSV Zip file to UCM and have access to the UCM account scm/item/import.

Perform these steps to upload to the UCM:

1. From the Navigator link, select the File Import and Export action link under the Tools region.
2. On the File Import and Export page, click the Create icon in the Search Results table header.
   a. Click on the Browse button and select the Zip file that you created for the import.
   b. Select the account from the choice list. The account used for item import is scm/item/import.
   c. Click the Save and Close button to initiate the uploading of the file.

**Import Data from the Item Management Interface Tables: Explained**

Once the CSV file is uploaded to UCM, you use the Load Interface File for Import scheduled process to move the data from the UCM folder to the interface tables.

Perform these steps to move the data into the interface tables:

1. From the Navigator link, select the **Scheduled Process** link.
2. Click the **Scheduled New Process** button in the Search Result table header.
3. In the Schedule New Process dialog select the following:
   a. **Type**: Job
   b. **Name**: Load Interface File for Import
4. Click the **OK** button.
5. In the Process Details dialog, enter the following parameters:
   a. **Import Process**: Item Import
   b. **Data File**: the name of the ZIP file you uploaded to UCM
6. Click the **Submit** button to start the process.
7. Monitor the status of the scheduled process for moving the data from the Item Management interface tables to the Item Management production tables by entering the process ID in the Search region in the scheduled processes user interface.

**Import the Data from Item Management Interface Tables**

At this point in the import process, the data is now loaded into the Item Management interface tables and is ready to be imported into the Item Management production tables. The next process will perform the item import process that will validate the data for import and move the data into the Item Management production tables. After successful completion of this job, the data will be available in the user interface.

1. From the Navigator link, select the **Scheduled Processes** action in the Tools section.
2. Click the **Schedule New Process** button in the Search Result table header.
3. In the Schedule New Process dialog select the following:
   - **Type**: Job
   - **Name**: Item Import

4. Click the **OK** button.

5. In the Process Details dialog, enter item import for the import process.

6. Enter the following information in the Process Details dialog:
   - **Batch ID**: enter a unique identifier of the batch used to import the items.
   - **Process All Organizations**: Select Yes to import items from all organizations. Select No to not import items from all organizations.
   - **Process All Organizations**: Select Yes to import items from all organizations. Selecting No to not import items from all organizations.
   - **Process only**: Indicate which transaction-type records need to be processed for an item batch. Values are Create, Sync, Update.
   - **Delete Processed Rows**: Indicates if the rows in the interface tables are to be deleted after the processing of an item batch. Values are Yes or No.

7. Click the **Submit** button to start the process.

8. After you click the Submit button a Configuration dialog is launched with the process ID. Note the ID.

9. Multiple processes are created for all of the steps in importing the item. The child processes are: Item Import Preprocessing, Item Import Data Quality, and Item Import.

10. View the log file for any errors (if applicable), by selecting the row in the Search Results table and clicking the **View Log** button.

Successfully imported items and related child entities can be queried, modified and validated.

**Monitor Item Imports: Explained**

The Monitor Item Imports task is used to monitor the Item Import process.

Navigate to the Monitor Item Imports task in the Setup and Maintenance work area, to search for specific Enterprise Service Scheduler processes and monitor their status in the search results table.

**Related Item Subtypes: Explained**

A related item is an item relationship between two existing items. How the two items are related is defined by a subtype.

Multiple subtypes for related items are seeded, and you can define additional subtypes using the Manage Related Item Subtypes task.

Seeded values are:

- Accessories
- Collateral
- Complimentary
- Conflict
- Cross-Sell
- Fulfillment
- Impact
- Mandatory Change
- Merge
- Migration
- Optional Change
- Option charge
- Prerequisite
- Promotional upgrade
- Repair to
- Service
- Split
- Substitute Supersed
- Upsell
- Warranty

Item Revision Descriptive Flexfields: Explained

Use item revision descriptive flexfields to capture item revision information whose values may differ between revisions of the same item.

Manage this flexfield type by using the Manage Item Revision Descriptive Flexfields task in the Setup and Maintenance work area.

Trading Partner Item Descriptive Flexfields: Explained

When defining descriptive flexfields associated with trading partner items, you must use certain prefixes when naming the context segments, in order for the segments to be displayed for the respective trading partner type.

The prefixes required for naming the context segments are listed in the following table, with their corresponding trading partner types. For example, if you define a trading partner item descriptive flexfield with a context segment named COMP_TPI_ATTRIBUTES, then the value segments of this context will be displayed for Competitor Item when users conduct transactions in that context.

<table>
<thead>
<tr>
<th>Trading Partner Type</th>
<th>Prefix for Context Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Competitor Item</td>
<td>COMP</td>
</tr>
<tr>
<td>Customer Item</td>
<td>CUST</td>
</tr>
</tbody>
</table>
Trading Partner Type | Prefix for Context Segment
---|---
Manufacturer Item | MFG
Supplier Item | SUPP

Manage this flexfield type by using the Manage Trading Partner Item Descriptive Flexfields task in the Setup and Maintenance work area.

Managing Document Classes: Explained

These are the steps for setting up and managing Document classes.

**Create Root Document Class**

- Use the Manage Item Classes setup task. This is found in Product Management > Items > Manage Item Classes.
- In the Create Item Class dialog, create a subclass of the root item class. You may want to call it Root Document Class, or a name that parallels the name of your root item class.

**Assign Root Document Class**

- Use Setup in the Product Development workspace.
- Enable Document Management.
- Designate your root document class.

**Add Document Subclass**

- Use the Manage Item Classes setup task to add subclass to the root document class.
- Users will be able to select from those subclass when creating a document object.
Define Catalogs

Catalogs: Overview

Before you can create catalogs and associate items with them you must complete several tasks in the Setup and Maintenance work area:

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Catalog Descriptive Flexfields</td>
<td>Descriptive flexfields can only have one context available at a single time.</td>
</tr>
<tr>
<td>Manage Category Descriptive Flexfields</td>
<td>Descriptive flexfields can only have one context available at a single time.</td>
</tr>
<tr>
<td>Manage Functional Area Catalogs</td>
<td>Each functional area can have a set of rules to define how a catalog should be configured to support the functional area. When a catalog is created and assigned to the functional area, it is validated against the functional area rules. For example, many of the Supply Chain Management applications participate in a process which automatically assigns an item being created to the default category in a functional catalog. Also if no catalog is assigned to the functional area, the functional area is skipped in the automatic assignment process. This task is only necessary if catalogs are to be used with other SCM products such as procurement or inventory.</td>
</tr>
</tbody>
</table>

Catalog Descriptive Flexfields: Explained

Descriptive flexfields are available at the catalog level to allow the user to define attributes for catalogs. For example, a customer wants to add attributes to the catalog to identify the usage of the catalog in their business process. An attribute called Usage is created as a descriptive flexfield for the catalog. The values are defined indicating what processes the catalog data is used in, such as new product development.

You create descriptive flexfields using the Manage Catalog Descriptive Flexfields task in the Setup and Maintenance work area.

Category Descriptive Flexfields: Explained

Descriptive flexfields are available at the category level to allow the user to define attributes for category in all catalogs where it is used.

For example, a customer wants to add attributes to the category to identify the packaging types for the items assigned to the category. An attribute called Packaging Type is created as a descriptive flexfield for the catalog. The values for this flexfield indicate what the packaging type is, such as box or case.
You create descriptive flexfields using the Manage Category Descriptive Flexfields task in the Setup and Maintenance work area.

Create Catalog: Explained

Item catalogs provide a mechanism to classify or group a set of items together based on common meaning. Catalogs can have a flat or single-level structure of categories or have a hierarchical structure categories.

For example, the item catalog Engine describes a group of categories that make up an engine such as engine block, carburetor, or ignition.

The items are assigned to the categories and represent components that make up the part of the engine. For example, spark plugs are a component of the ignition category.

Catalogs can be hierarchical and can contain a hierarchy where parent and child relationships between the category are used for classification, or a catalog can contain only one level, no hierarchy where the catalog is a list of categories.

For example, a category can be configured to be a browsing category by configuring the category to allow only allow categories to be added. In addition you can configure the category to allow both categories to be associated to it in a hierarchy and items can be assigned to it, as in the case where the category parent category in the hierarchy.

Item catalogs have two types: functional area catalogs that are created and maintained through the Manage Functional Area Catalogs task in the Setup and Maintenance work area and Product catalogs that are created and maintained in the Production Information Management work area. Product Hub catalogs cannot be assigned to a functional area.

Functional area catalogs are primarily used to support other Oracle Fusion applications that require integration between the functional area catalog and the process within the application. For example the Purchasing functional area catalog is integrated with the Procurement processes to allow the items assigned to the categories in this catalog to be used to support the purchasing processes.

Product catalogs are used to support additional processes and integration with external applications.

Create functional area catalogs using the Manage Functional Area Catalogs task in the Setup and Maintenance work area:

1. In the Navigator, click Setup and Maintenance.
2. On the Setup and Maintenance page, click the Manufacturing and Supply Chain Materials Management offering, and then click Setup.
3. On the Setup: Manufacturing and Supply Chain Materials Management page, click the Catalogs functional area, and then click the Manage Functional Catalogs task.
4. Click the Create icon.

Create product catalogs using the Manage Catalogs task in the Product Information Management work area:

1. Launch the Manage Catalogs task in the Product Information Management work area.
2. Create the catalog using the Create icon.

Manage Catalogs: Explained

You can edit a catalog after it has been created, using the Manage Functional Area Catalogs task in the Setup and Maintenance work area. You can also access this task from the Product Information Management work area, if you have Oracle Fusion Product Hub installed.
To edit functional area catalogs using the Manage Functional Area Catalogs task in the Setup and Maintenance work area:

1. In the Navigator, click Setup and Maintenance.
2. On the Setup and Maintenance page, click the Manufacturing and Supply Chain Materials Management offering, and then click Setup.
3. On the Setup: Manufacturing and Supply Chain Materials Management page, click the Catalogs functional area, and then click the Manage Functional Area Catalogs task.
4. Search for the catalog.
5. Click the link in the Catalog Name column or select the row and click the Edit icon.

Create Product Hub catalogs using the Manage Catalogs task in the Product Information Management work area.

1. Launch the Manage Catalogs task in the Product Information Management work area.
2. Create the catalog using the Create icon.

Edit Product Hub catalogs using the Manage Catalogs task in the Product Information Management work area.

1. Search for the catalog on the Manage Catalogs search page.
2. Click the link in the Catalog Name column or select the row and click the edit icon.

Category Hierarchy Tab

This contains the category hierarchy region in which the category hierarchy can be created and maintained. In addition, items can be assigned, the usage of the category in other catalog can be viewed, and the attributes for the category and catalog category association can be edited.

This tab also provides an action to allow the category hierarchy to be edited in a spreadsheet or a complete hierarchy to be edited. For example, a customer may be using a UNSPC classification. They can download the spreadsheet to their desktop and cut and paste the UNSPC classification hierarchy into the spreadsheet and upload the spreadsheet to the application.

Selecting a category will open the category detail region. This regions contains three additional tabs for the item category assignments, category details and category attachments.

Catalog Detail Tab

This tab contains contains the catalog name and description, an image, the selection of the default category, the start and end date for the catalog and the catalog descriptive flexfields.

The default category is used by the automatic item assignment process that is run when:

- Category is assigned to the default category field in the catalog header. During automatic assignment process the item is assigned to the default category that is referenced by this field, for the catalog assigned to the functional area.
- Attribute values specified in the rules are met.

The start and end date allow the catalog life cycle to be controlled.

The Public Catalog check box is used to enable data security for catalogs and categories.

Catalog Attachments

The Attachments tab is used to add attachments related to the catalog such as related documents or images.
Functional Area Catalogs: Explained

Functional areas represent products or functionality of the product.

Each functional area can have a set of rules to define how a catalog should be configured to support the functional area. When a catalog is created and assigned to the functional area, it is validated against the functional area rules. For example, many of the Supply Chain Management applications participate in a new item process which automatically assigns an item being created to the default category in a functional catalog. Also if no catalog is assigned to the functional area, the functional area is skipped in the automatic assignment process.

Automatic Assignment Catalogs: Explained

The automatic assignment catalog feature is a simple way to create a non-hierarchical catalog because you do not have to add categories manually to the catalog. This feature adds the categories at the root level, so it works with both flat and hierarchical catalogs.

All categories that have the same category structure value as the catalog are automatically assigned and associated to the catalog when you create a catalog category association for each category.

Automatic Assignments

The automatic assignment feature is enabled during catalog creation when you select the Enable automatic assignment of category check box. The categories displayed for auto assignment catalogs are refreshed only at start up and after you save.

Note that if you create a category in another catalog with the same structure value as the automatic assignment catalog, the category is also added to your catalog. The categories displayed for auto assignment catalogs are refreshed only at start up and after you save.

When you open a new catalog, any categories that have the same category structure value as the catalog structure value for the catalog are automatically assigned to the catalog.

For example, Purchasing may maintain a master catalog containing all categories that represent commodities. Each commodity team can create categories for their commodity in their own catalog.

The master catalog for purchasing is named Purchasing and is configured during creation to support the automatic assignment of categories. Because you enabled automatic assignments for the Purchasing catalog, any categories created by the commodity teams are added to the catalog automatically. The purchasing managers can view the collection of all commodities represented as categories in the Purchasing catalog.

Catalog Report Publishing: Explained

Other applications can use catalog data if you export the catalog content. For example, you may want to export catalog content to use as a monthly report of all items assigned to a specific catalog. You can use the default publish template provided in hyper text markup language (HTML). You can specify the content and layout of the catalog information. When the catalog is published, you select the format and initiate the creation of the content in the file.
Publish a Catalog

Search for a catalog from the Manage Catalogs page, select the row corresponding to the catalog that you want to publish and select the Publish action. The application generates the report based on the default template in HTML format. You can select a new template or format from the report window. The content displayed for items, categories, catalog categories, and catalog is based on the publish template. The seeded template is called Catalog Listing. The template controls what data is in the report and how it is formatted.

Type of Catalog Content That Can Be Published

The default catalog publish template allows the publication of the catalog header details, category hierarchy, category details, and category item assignments. The order of a published report begins with the catalog header and the catalog category details. If the category has a child relationship, then the catalog category association details for the child category follows. If the child category has a hierarchy, then the complete hierarchy under the category is published with the catalog category association details and categories details.
5 Define Change Orders

Change Order Setup: Overview

Before you can create change orders, you must complete these tasks in the Setup and Maintenance work area under the Product Management offering:

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Change Reasons</td>
<td>Change reasons are used to categorize and establish the cause or reason for a change. Change reasons are required for creating change orders.</td>
</tr>
<tr>
<td>Manage Change Priorities</td>
<td>Change order priorities are used to identify the criticality of changes. Change order priorities are required for creating change orders.</td>
</tr>
<tr>
<td>Manage Change Statuses</td>
<td>Change statuses enable you to manage a change order through its life cycle.</td>
</tr>
<tr>
<td>Manage Change Order Types</td>
<td>All change orders are assigned a change order type. You must define at least one change order type to use change orders.</td>
</tr>
<tr>
<td>Manage Change Order and New Item Request Header Descriptive Flexfields</td>
<td>Used to manage the header-level descriptive flexfields for change orders and new item requests.</td>
</tr>
<tr>
<td>Manage Change Order and New Item Request Line Descriptive Flexfields</td>
<td>Used to manage the line-level descriptive flexfields for change orders and new item requests.</td>
</tr>
<tr>
<td>Manage Organization Trees</td>
<td>Organization trees are used to create a list of organizations for use in some of the mass change flows. These are optional.</td>
</tr>
<tr>
<td>Manage Task Configurations for Supply Chain Management</td>
<td>Define the task configurations for the change order approval workflow.</td>
</tr>
<tr>
<td>Manage Approval Groups for Supply Chain Management</td>
<td>Define approval groups for the change order approval workflow.</td>
</tr>
</tbody>
</table>

Change Order Reasons: Explained

Change reasons are used to categorize and establish the cause or reason for a change. Change reasons are required for creating change orders.

Processes and rules can be defined around change reasons so that change orders with safety related reasons get priority over other change orders or they get routed through different levels of approvals.

The predefined values for change reasons are: Cost, Quality, and Safety
Additional values can be added, modified or deleted by accessing the Manage Change Reasons task in the Setup and Maintenance work area.

**Change Order Priorities: Explained**

Change order priorities are used to identify the criticality of changes. Change order priorities are required for creating change orders.

Processes and rules can be defined around change priorities so that change orders with a high priority get priority over other change orders or they get routed through different levels of approvals.

The predefined values for change priorities: High, Medium, and Low

Additional values can be added, modified or deleted by accessing the Manage Change Priorities task in the Setup and Maintenance work area.

**Change Order Statuses: Explained**

Change statuses enable you to manage a change order through its workflow.

You manage the progression of a change order through its workflow by promoting it (or sometimes demoting it) to the next in a series of change statuses. For each change type, you define a set of applicable statuses.

The change status types are:

- Open
- Interim Approval
- Approval
- Scheduled
- Completed

As an administrator, you can create statuses of each change type and assign a name. You can apply different statuses to each change order type to form workflows unique to each status type.

If you use autopromote and autodemote in the same workflow, ensure that you do not autodemote the change order to the same status from which it was automatically promoted. For example, in a workflow consisting of Open, Approval, Scheduled, and Completed status:

Do not autopromote from Open to Approval and do not autodemote from Approval to Open (when the approval is rejected).

It is recommended that in the same workflow, create an additional status between Open and Approval as follows:

Open, Rework (type Open), Approval, Scheduled, and Completed.

Then, set autopromote and autodemote as follows:

- Autopromote from Open to Approval
- Autodemote from Approval to Rework (when the approval is rejected)
Draft Status
By default, the Draft status is the first status for all change orders irrespective of the change order type. You cannot configure Draft and it does not appear in the workflow. In this status, you can modify the change order.

Open Status
In Open status, you can make the following changes to the change order: add specific items and modify attribute values, select priority and reason, and provide description and optional supporting documents. You can have more than one open status in the change order workflow. For example, Open, Interim Approval, Open, and Interim Approval.

Interim Approval Status
You can optionally add one or more Interim Approval status, to accommodate multiple approval flows or progressively approve change orders in your organization.

Approval Status
Change orders can be routed to a list of approvers. Approvers are defined in change order type, or if the approval is rule-based then approvers are defined in approval groups. Depending on the type definition, more than one person can approve. If the approval is rule-based, then approvers are defined in approval groups in the BPM worklist and approval groups are associated with rules. Only user-defined approvers are defined at the type level. In a user-defined workflow, approvers can also be added at run time by either the initiator or the Assigned To person.

Scheduled Status
When a change order is approved, it is automatically promoted to a Scheduled status. After the change order is scheduled, it cannot be demoted or canceled. The change order remains scheduled until all the affected objects have reached their effective date.

Completed Status
When the effective date of items in the change order is reached, the changes defined in the change order become effective in production. When all item lines in the change order are effective, the change order is completed. The change order cannot be reopened or canceled once in this status.

Change Order Types: Explained
All change orders are assigned a change order type that defines the attributes and workflow of the change order. A change order type can be end-dated if it is not used in any change orders that have a workflow in progress. You must define at least one change order type to use change orders. Use the Manage Change Order Types task (in the Setup and Maintenance work area) to create and modify change order types.

This table provides information on change order types and supported features.
<table>
<thead>
<tr>
<th>Change Order Type</th>
<th>Description</th>
<th>Supports Revision Control</th>
<th>Supports Audit Report</th>
<th>Supports Redline Modifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineering Change Order</td>
<td>Used when tracking major changes during the design phase of the item.</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes.</td>
</tr>
<tr>
<td></td>
<td>Displays the following modifications in redline: item detail, EFF, structure, AML and attachment.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Order without Revision Control</td>
<td>Used when revision control is not required or when changes are minimum. For example, replacing an existing manufacturer part with a similar part supplied by another manufacturer.</td>
<td>No</td>
<td>Yes</td>
<td>Yes.</td>
</tr>
<tr>
<td></td>
<td>Displays the following modifications in redline: AML, EFF, and item detail.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Request</td>
<td>Used when suggesting changes for a released item. For example, a supplier could request changes in certain specifications.</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Deviation Change Request</td>
<td>Used when deviating from a process or specification for a specific time period. For example, suggesting the use of a substitute component as a temporary means of resolving an issue.</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Commercialization Change Order</td>
<td>Available only in the Product Hub work area.</td>
<td>Yes</td>
<td>No</td>
<td>Yes.</td>
</tr>
</tbody>
</table>

The change order type contains the following information:

- Default values for Assigned To and Item Effective Date
- Number generation method
- Entry and exit criteria, and workflow set up
- Propagation rules

**Note:** Changes made to a change order type will not be applied to any existing change orders for this type.

The following fields appear as part of the Manage Change Order Types task:
Assigned To
Any change order created of the particular change type would be assigned by default to this user or group. You can modify this value when the change order is created. This user is like a Change Analyst who is notified about status changes and approvals to keep the change order on track.

Item Effective Date
Specify the default number of days, after the change order’s creation date, on which you want the item changes to become effective. Alternatively, specify that the changes become effective immediately when the change order is completely approved. This option sets the default item effective date when a change order is created, which can be modified before the change order is submitted. If the effective date for an item is empty, the item change becomes effective immediately upon approval of the change order.

> Note: It is the effective date of the specific items in a change order that determines their production effectivity, not the item effective date for the change order, which is used to set the default effective date of affected object (or change line) in the change order.

Number Generation Method
When the administrator selects Sequence Generated, the administrator can provide the Prefix, Starting Number, Increment, and Suffix for change order numbers for the change order type.

When Rules Generated is selected as a number generation method, the user needs to associate a user-configured rule set, which creates change numbers in the sequence defined in the function.

You can also use the User-Defined method and define your own number generation methods.

Entry and Exit Criteria
Entry and exit criteria are set up as rules that validate progression of a change order through its workflow. You define entry and exit criteria for each workflow status to serve as checkpoints in a change process flow.

Entry criteria can be set up for Interim Approval and Approval status. Exit criteria can be set up for Open and Interim Approval status.

Before you create entry and exit criteria, create a change order type. Associate the change order type with the entry and exit criteria. Again associate the entry and exit criteria with the change order type (by using the Manage Change Order Entry and Exit Criteria task or the Manage Item Rule Sets task in the Product Management offering).

Create validation rules by using the Manage Item Rule Sets task in the Product Management offering (available in the Setup and Maintenance work area); select association type as Change order type. Edit the change order type to select the entry and exit criteria (in the Workflow tab).

The following attributes may be used to create the validation rules:

- Change Header
  - Priority
  - Reason
  - Need-by Date
Associating Propagation Rules to Change Order Types

If propagation rules have been configured in the change order type, then the propagation organization appears in the change order header, the default list of organizations, or organization hierarchy identified by the change order type. You can select other organizations or organization hierarchies that have been identified in the change order type. The list of organizations associated with the selected propagation organization or hierarchy appear in the Propagation Organizations region. Note that propagation rules are only applicable in the Product Hub work area.

Propagation rules are associated with the change order type that you select for the new change order. You define propagation rules when you create a change order type or edit an existing one on the Propagation Rules tab.

Specify the organizations from which a change order might be propagated. For each specified source organization, select one or more target organizations or organization hierarchies where the change order can be propagated.

Each change order type can be configured to support propagation from different organizations and propagation to different organizations or organization hierarchies.

Manage Change Order Entry and Exit Rule Sets: Explained

This topic describes how to manage Change Order entry and exit criteria.

You can configure what fields must appear as Required Fields when a change order enters or exits a workflow state, by using either of the following tasks:

- Manage Change Order Entry and Exit Rule Sets (in the Setup and Maintenance work area)
- Manage Item Rule Sets (in the Setup and Maintenance work area and the Product Management offering)

This could be any field in the change object’s general information page, or extensible flexfield for general information page and affected items. This could also include fields on the item object or item BOM/AML. For example, a rule set that enforces need-by date value to be mandatory for high priority change orders.

Using the Manage Item Rule Sets task, you can also enforce the value of a change order attribute based on another attribute.

Examples:

- If the user selects Reason as Quality, then the priority of change order can be enforced as High. If the user selects the priority as Low, then change order can be restricted from progressing to the next status.
- A rule set specifies that when a Change moves from Pending to Submitted state, all items in the Engineering Change Order (ECO) should have a description filled out. To ensure this, mark the Description field as a required field.
For each rule set, use the **Type** options to filter the fields by type. Select the fields that should appear as required fields. In addition to setting up criteria for the whole workflow, you can specify criteria for a specific status within the workflow.

- You can determine required fields based on **Change** and **Affected Item** attributes. For example, if **Requires Implementation Plan** is **Yes**, ensure text is entered in a field called **Implementation Plan**.

**Related Topics**

- Defining Product Development: Overview
- Product Development Components: How They’re Configured

**Data Security Privilege for Viewing Change Orders: Explained**

By default, the product manager role provides data security privilege for viewing change orders. If you have modified that role or if a user is unable to view the change order, create a data security policy for the user.

To create a data security policy, follow these steps:

1. Sign in with your Oracle Fusion user ID and password.
2. Select **Navigator > Security Console**.
3. Search for the duty role named **ORA_EGP_PRODUCT_COMMON_MGT_DUTY**.
4. If none of the roles include **ORA_EGP_PRODUCT_COMMON_MGT_DUTY**, assign this to a role.
5. Edit the role from which **ORA_EGP_PRODUCT_COMMON_MGT_DUTY** is inherited.
6. In data security policy, click **Create New Policy**.
7. In data set, choose the instance set option.
8. Select the condition name and the actions associated with the data security policy and click **OK**.
9. Click **Next** and save your changes.
6 Define Product Rules

Product Rules: Overview

Before you can use product rules in Oracle Fusion Product Hub, you must complete this task in the Setup and Maintenance work area.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Item Rule Sets</td>
<td>Use this task to create validation, assignment or composite rule sets.</td>
</tr>
</tbody>
</table>

Rules and Rule Sets: Explained

Rules define integrity constraints on the attributes of items and structures. You can define integrity constraints on operational as well as on user-defined attributes. Integrity constraints often implement business rules and are created through use of the rules framework. For example, a rule might be that the minimum speed must be less than maximum speed.

Rule sets gather multiple rules together and are assigned to an association type, such as attribute groups, item classes, change types, or structure types. They also list valid business entities. For example, an item, item supplier, or item revision. This enables the expressions entered to be validated by checking for allowable entities.

Keep in mind that:

- If the rule set is assigned to an attribute group, then only the attributes in that group can be used in the rule.
- If the rule set is assigned to an item class, then only the attribute groups valid for that item class will be used.

The status of a rule set can be identified as draft. Users can keep a rule set in draft status until the drafting of rules is complete. If so, the rule set is not triggered as regular transactions are completed. During this time, simulation analysis can be run to study the impact of the rule sets on a selected set of existing items, enabling users to make necessary changes. While performing the simulation, the draft rule sets along with other active rule sets are applied on the selected set of items, and the impact is captured; this is handled by an asynchronous scheduled process.

The rules for new item request and change order type associations are used to generate new item request and change order numbers.

The types of rule sets and rules are as follows:

- Validation
- Assignment
- Composite
Validation

Validation rule sets validate conditions based on attribute values defined for items. They are typically used to model predefined business rules on items.

Each attribute is referenced by its business entity and attribute group name followed by the attribute name. For example, item.[Item].[Physical Attributes].[Unit Volume]. In this example, [Item] indicates that it is an item attribute; [Physical Attributes] is the display name of the attribute group, and [Unit Volume] is the display name of the attribute.

Validation rules restrict items that can be added as related items to an item, and restrict the relationship types that can be allowed for items. This restriction could be based on item or item revision-level attributes which could be operational or EFF.

Test the validations by going to the Item Update page and editing the appropriate attribute groups. Updated values are validated against the rules, and error messages appear on the screen.

Assignment

An assignment rule set determines the value of an item attribute based on the specified condition. Generally, rule sets for assignments should be executed before validations, since they permit you to write validation conditions ensuring that the result of the assignments are valid.

An example of an assignment rule is, Lead Percent is Total Lead Mass divided by Unit Weight.

After you create a rule, you validate and save it. Then, if necessary, enter subsequent rules. Rules are executed in the order of their sequence in the rule set. Therefore, if an attribute’s expression depends on a previously calculated value, you must ensure that the previous value appears ahead of the attribute and is therefore computed first.

Composite

Composite rules set can contain both validation and assignment rule sets. They can be used to aggregate rule sets that operate on different attribute groups and item classes.

You create a composite rule set on the Manage Rule Sets page. To define a composite rule set of mixed type, ensure it contains both validation and assignment rule sets. Set the type to Mixed, enabling the creation of a rule set that contains both assignment and validation rule sets. Then add assignment and validation rule sets to the composite rule set.

Related Topics

- What’s a rule set?
- What are item rules?
- Generating Numbers and Descriptions with Rules: Examples

Item Rule Data Types: Explained

Attributes in item rules belong to one of a set of data types.

In item rule expressions, all values are of one of the data types listed in the following table.
### Data Type

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>All strings and text characters</td>
</tr>
<tr>
<td>Number</td>
<td>All decimals and integers</td>
</tr>
<tr>
<td>Date</td>
<td>Dates only, without times</td>
</tr>
<tr>
<td>Time</td>
<td>Times only, without dates</td>
</tr>
<tr>
<td>DateTime</td>
<td>Combination of Date and Time</td>
</tr>
</tbody>
</table>

### Null or Empty Values

Any attribute referenced in an item rule can have an empty or null value. To test for empty or null values you must use the function `isnull`.

### Strings

Strings are delimited by double or single quotation marks (`"` or `'`). To escape a character’s normal value (for example, in a regular expression), prefix it with the backslash (`\`).

### Numbers

Depending on the setting of the regional preference for number format, the decimal point in decimal numbers can be represented by either a period character (.) or by a comma (,). If the comma is used as a decimal point, then arguments to functions must be delimited by semicolons (;).

### Dates and Times

Dates and times are represented by the types Date, DateTime, and Time.

Dates will be entered in the format that you specify in the setting of the regional preference for date format. Date constants will automatically be reformatted if you log in with a different preference.

The following expression using a Date value is valid:

```
[Item].[Logistics].[StartDate] > "2005-Feb-23"
```

When writing an expression containing terms representing dates and times, enclose them in double quotation marks. They are internally converted to String values before they are compared. The following expression compares an attribute value to a textual date term.

```
[Item].[Logistics].[SellDate] > [Logistics].[AvailableDate] + 30
```

Days can be added to a Date using Numbers. For example, the following expression computes the sell date as 30 days after the availability date:

```
[Item].[Logistics].[SellDate] > [Logistics].[AvailableDate] + 30
```

You cannot perform date calculations involving Custom Calendars.
Item Rule Syntax: Explained

Item Rules are used in defining and validating integrity constraints on item attributes, and in assigning values to attributes.

Basic Rule Formation

Item rules consist of expressions, operators, and functions. Most expressions reference the value of an item attribute.

Rules can encapsulate a single integrity constraint. The following example checks that a given attribute is less than another:

\[
[\text{Item}].[\text{Physical Attributes}].[\text{Unit Weight}] \leq [\text{Item}].[\text{Logistics}].[\text{Shipping Weight}]
\]

In other words, the net weight of an item always has to be less than or equal to the shipping weight.

Rules can define assignment formulas. The following example is used in assigning a value for a "Daily Waste Percent" attribute:

\[
[\text{Item}].[\text{Logistics}].[\text{Total Waste Percent}] / [\text{Item}].[\text{Inventory}].[\text{Shelf Life Days}]
\]

In the preceding example, the daily waste percent is the total waste percent divided by the shelf life in days.

Attribute Expressions

You access the value of an attribute by appending its name to its entity and attribute group, separated by a delimiter. The delimiter is the period character (.)

Following is the syntax of an attribute expression.

\[
[<\text{Entity Name}>].[<\text{Attribute Group Name}>].[<\text{Attribute Name}>]
\]

When referencing descriptive flexfields, use the segment code.

\[
[\text{ChangeHeader}].\text{Flexfield}[\text{Product__Line}]
\]

Null Values

Rules that reference attributes that have no value (also called a null value) are ignored. Expressions that evaluate to NULL are ignored.

That means that you never need to code Validations like:

\[
\text{if (not isnull(Item.PhysicalAttributes.Weight)) then Item.PhysicalAttributes.Weight} \leq 10
\]

The "if" part is redundant, since if Weight was NULL the validation would be ignored. If you do want to check that an attribute has a non-null value, use the \text{isnull} function, which can be used to check that an attribute has a value entered.

Boolean Expressions

Boolean expressions are those that return TRUE, FALSE or null. Boolean expressions can be used in \text{If Expression} and \text{Validation Condition} fields of the Edit Rule Set page. Use logical and comparison operators and functions in Boolean expressions.
Item Rule Multirow Attribute Group Functions

To reference a value in a particular row of a multirow attribute group, use one of the following functions.

To make a rule set specific to an item class, assign it to that item class during rule set creation. That rule set will then be active for each entity that has that item class as a parent or as an ancestor. You cannot access any other attributes associated with an item class.

**loopSum**

Syntax:

```
loopSum(([Entity name].[multirow Attribute Group Name].[Attribute Name])
```

The loopSum function takes one numeric sub-expression as an argument. It runs the sub-expression for each multirow row attribute group and compute the sum of the results.

Example

In this example, the sum of the values in all the rows of Percentage attribute cannot be a value other than 100.

```
Severity: Reject
If Expression: (loopSum([Item].[Composition].[Percentage])) != 100
```

**conditionalLoopSum**

Syntax:

```
conditionalLoopSum(boolean_expression, [Entity name].[multirow AttributeGroupName].[Attribute Name])
```

The conditionalLoopSum function takes two sub-expression arguments. The first argument must be a boolean expression and the second argument will be a numeric sub-expression. The function runs the boolean sub-expression for each multirow row and, if the boolean evaluates to true, computes the numeric sub-expression for that row. The function returns the sum of the computed numeric expressions.

Example

In this example, assume a multirow attribute group named `Forecast` with the following rows.

<table>
<thead>
<tr>
<th>Customer</th>
<th>Location</th>
<th>Required Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td>Seattle</td>
<td>20</td>
</tr>
<tr>
<td>XYZ</td>
<td>Seattle</td>
<td>30</td>
</tr>
<tr>
<td>ABC</td>
<td>Boston</td>
<td>25</td>
</tr>
</tbody>
</table>

The following example will sum up the values of Required Quantity for which the corresponding Location is Seattle:

```
conditionalLoopSum([Item].[Forecast].[Location] == "Seattle", [Item].[Forecast].[Required Qty])
```
Item Rule Numeric Functions and Operators

Use numeric functions and operators to calculate values of numeric expressions.

Numeric Operators
The numeric operators available in item rules are listed below.

- + (addition)
- - (subtraction)
- * (multiplication)
- / (division)
- sum (aggregation)

Numeric Operator Expressions

expression1 + expression2
Performs regular mathematical addition of expression1 and expression2. Returns null if any argument is null.

expression1 - expression2
Performs regular mathematical subtraction of expression2 from expression1. Returns null if any argument is null.

expression1 * expression2
Performs regular mathematical multiplication of expression1 and expression2. Returns null if any argument is null.

expression1 / expression2
Performs regular mathematical division of expression1 by expression2. Returns null if any argument is null. Division by 0 returns null. The number of decimal digits returned by division is the maximum number of digits from expression1 and expression2.

sum(expression1, expression2, expression3, ...)
Adds a series of values. Returns null if any argument is null.

Numeric Functions
The numeric functions available in item rules are listed below.

abs
Syntax:
  abs (expression)
Returns the absolute value of expression.
Example:
In this example, the percentage weight change has to be less than 10.
\[
\text{abs(percent([Item].[Physical Attributes].[Unit Weight]))} \leq 10
\]

**ConvertToUOM**

Syntax:
\[
\text{ConvertToUOM(expression, "target UOM")}
\]

This conversion ensures that comparisons or calculations are performed using the appropriate UOM.

Example:
In this example, the unit weight of an item has to be less than or equal to 10 kg.
\[
\text{ConvertToUOM([Item].[Physical Attributes].[Unit Weight], "kg")} \leq 10
\]

**max**

Syntax:
\[
\text{max(expression1, expression2, ...)}
\]

Example:
In this example, the result would be 2006-11-30.
\[
\text{max("2006-10-12", "2006-11-30")}
\]

**min**

Syntax:
\[
\text{min(expression1, expression2, ...)}
\]

Example:
In this example, the result would be 2006-10-12.
\[
\text{min("2006-10-12", "2006-11-30")}
\]

**round**

Syntax:
\[
\text{round(expression, decimal_places)}
\]

Example:
In this example, the returned value is 1.58.
\[
\text{round(1.5758, 2)}
\]
rounddown
Syntax:
rounddown(expression, decimal_places)
Rounds a number to the specified number of decimal places. rounddown rounds toward zero.
Example:
In this example, the returned value is 1.57.
rounddown(1.5758, 2)

roundup
Syntax:
roundup(expression, decimal_places)
Rounds a number to the specified number of decimal places. roundup rounds away from zero.
Example:
In this example, the returned value is 1.58.
roundup(1.5758, 2)

Item Rule Production Value Functions

Use item rule production value functions and operators to compare the new value of an attribute to the value that is currently in production. This can be especially useful in tolerance rules.

changed
Syntax:
changed(attribute)
changed(attributeGroup)
Returns TRUE if the current value of attribute or attributeGroup differs from the current production value. Returns FALSE otherwise. Works with null values. If only attributeGroup is specified, then returns TRUE if any attribute in that attribute group has changed.

delta
Syntax:
delta(attribute)
Returns the difference between new and current production values of attribute. Comparisons of String values are case-insensitive. In comparisons of Boolean values, TRUE is considered greater than FALSE.

The following table illustrates the return value of this function when comparing new and current production values of various combinations of Numeric, Date, and DateTime attribute types.
### Value comparison

<table>
<thead>
<tr>
<th>Value comparison</th>
<th>Returned result for Numeric values</th>
<th>Returned result for Date or DateTime values</th>
</tr>
</thead>
<tbody>
<tr>
<td>new &lt; current production</td>
<td>new minus current production</td>
<td>new minus current production</td>
</tr>
<tr>
<td>new &gt; current production</td>
<td>new minus current production</td>
<td>new minus current production</td>
</tr>
<tr>
<td>new == current production</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>current production does not exist</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td>both new and current production are null</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>either new or current production are null, but not both</td>
<td>null</td>
<td>null</td>
</tr>
</tbody>
</table>

### percent

**Syntax:**

```plaintext
percent(attribute)
```

Returns the difference between new and current production values, expressed as a percentage, according to the following formula:

\[
\text{percent} = \left( \frac{\delta(\text{attribute})}{\text{current_production_value}} \right) \times 100
\]

Where `attribute` is the argument passed to `percent()`, and `current_production_value` is the current production value of the attribute. Can only be used with Numeric attributes.

### previous

**Syntax:**

```plaintext
previous(attribute)
```

Returns the previous production value of `attribute`.

### Item Rule String Functions

All string functions are case-sensitive. To perform a case-insensitive comparison use the `==` comparison operator.

#### String Comparison Functions

**compare**

**Syntax:**

```plaintext
compare(string1, string2)
```

Returns 0 when `string1` is exactly equal to `string2`. Returns -1 if `string1` is lexicographically less than `string2`. Returns +1 if `string1` is lexicographically greater than `string2`.
contains
Syntax:
contains(look_for_string, look_in_string)
Returns TRUE when look_for_string is found in look_in_string. Returns FALSE otherwise.
Example:
The following example returns TRUE if the item description is "computer accessory product".
If Expression: contains("accessory", [item].[main].[description])

endsWith
Syntax:
endsWith(look_for_string, look_in_string)
Returns TRUE when look_in_string ends with look_for_string. Returns FALSE otherwise.

match
Syntax:
match(regexp_pattern, look_in_string)
Returns TRUE when regexp_pattern matches look_in_string. Returns FALSE otherwise.
This function uses regular expression pattern matching in its search. For reference on regular expressions, see the Javadoc reference for java.util.regex.Pattern (http://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html).
Example:
The following example returns TRUE if the description includes "electronic" or "electric".
match("ele*c", [item].[main].[description])

startsWith
Syntax:
startsWith(look_for_string, look_in_string)
Returns TRUE when look_in_string starts with look_for_string. Returns FALSE otherwise.
Example:
The following example returns TRUE if the item description is "Notebook".
startsWith("Note", [item].[main].[description])

Other String Functions
+ (plus sign)
Syntax:
expression1 + expression2
Concatenates two expressions and returns the resulting string. Note that this also returns a valid string if the expressions are of other data types.

`indexOf`

Syntax:

```
(look_for_string, look_in_string)
```

Returns position of `look_for_string` in `look_in_string`. String position starts at 0. Returns -1 if `look_for_string` is not found. The search is case-sensitive. Returns null if either expression is null.

`length`

Syntax:

```
length(expression)
```

Returns the length of the string `expression`. Returns null if `expression` is null.

`lowercase`

Syntax:

```
lowercase(expression)
```

Returns the lowercase equivalent of the string `expression`. Returns null if `expression` is null.

`substring`

Syntax:

```
substring(string, start)
substring(string, start, end)
```

Returns a substring of the string `string` starting at `start` and ending before `end`. If `end` is omitted, then returns remainder of `string`. String position starts at 0. If `start` is less than 0 then start at the beginning of the string. If `start` is greater than length of `string` then return up to the end of the string. Returns null if any of the arguments is null.

Example:

You can use rules to validate that the Packaging Indicator digit for a GTIN is appropriate for the Pack Type of the item. GTINs can be assigned at multiple levels of a packaging hierarchy. Consider a scenario in which your GTIN numbering rule declares that, if the pack type of the Item is "EA", then the fourth digit of the GTIN should be 8. You can use the following rule expressions to perform this kind of validation.

```
If Expression: [Item].[Main].[Pack Type] == "EA"
Validation Condition: substring([GTIN].[GTINMain].[GTIN], 4, 4) == "8"
```

`trim`

Syntax:

```
trim(expression)
```

Removes all leading and trailing (but not middle) white space characters from a `expression`. Returns null if `expression` is null.
uppercase
Syntax:

    uppercase(expression)

Returns the uppercase equivalent of the string `expression`. Returns null if `expression` is null.

Item Rule Logical Functions and Operators

Use item rule logical functions and operators to test the validity of expressions.

Logical Operators

The logical operators available in item rules are listed below.

- **and** (logical AND)
- **or** (logical OR)
- **not** (logical NOT)

and

Syntax:

    expression1 and expression2

You can also use the notation `&&` in place of the `and` operator.

The logical `and` operator implements the following truth table for `expression1` and `expression2`.

<table>
<thead>
<tr>
<th><code>expression1</code></th>
<th><code>expression2</code></th>
<th><code>expression1 and expression2</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>null</td>
<td>F</td>
</tr>
<tr>
<td>null</td>
<td><em>[see note]</em></td>
<td>null</td>
</tr>
</tbody>
</table>

* The processor stops after it finds the first FALSE. Hence there is an asymmetry between `F and null` and `null and F`.

or

Syntax:
expression1 or expression2

You can also use the notation || in place of the or operator.

The logical or operator implements the following truth table for expression1 and expression2.

<table>
<thead>
<tr>
<th>expression1</th>
<th>expression2</th>
<th>expression1 or expression2</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>null</td>
<td>T</td>
</tr>
<tr>
<td>null</td>
<td>*[see note]</td>
<td>null</td>
</tr>
</tbody>
</table>

* The processor stops after it finds the first TRUE. Hence there is an asymmetry between T and null and null and T.

not

Syntax:

not expression1

The logical not operator implements the following truth table for expression1 and not expression1.

<table>
<thead>
<tr>
<th>expression1</th>
<th>not expression1</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>null</td>
<td>null</td>
</tr>
</tbody>
</table>

Logical Functions

The logical functions available in item rules are listed below.

assignedToOrg

Syntax:

assignedtoOrg("org_code")

In a validation expression, returns TRUE if the item is assigned to the specified organization.
Example:

In the following example, when the item class of the item is Exact Servers, the organization cannot be S2.

```
Severity: Reject
If Expression: [Item].[Item Basic].[Item Class] == "Exact Servers"
Validation Expression: !assignedtoOrg("S2")
```

**assignedtoCatalog**

Syntax:

```
assignedToCatalog(Catalog[CatalogCode].Category[CategoryCode])
```

In a validation expression, returns TRUE if the item is assigned to the specified catalog and category.

Example:

In the following example, if an item is assigned to the catalog Ladies Wear and the category Summer, it cannot be assigned to catalog Kids and category Summer.

```
Severity: Reject
If Expression:
assignedtoCatalog(Catalog[LadiesWear].Category[Summer])
Validation Expression:
!assignedtoCatalog(Catalog[Kids].Category[Summer])
```

**exists**

Syntax:

```
exists(Boolean_expression)
```

Loops through the rows of the entities used in `Boolean_expression` and returns TRUE if the expression is satisfied for any of the rows.

You can use `exists()` on the following entities:

- relationships (related item relationships and cross-references)
- structures
- multirow attribute groups

Example expressions:

The following expression loops through all the structures on an item and returns TRUE if the any structure has the name "ManufacturingBOM".

```
exists([Structure].[Structure Attributes].[Structure Name] == "ManufacturingBOM")
```

The following expression validates whether a particular relationship exists for an item:

```
exists([RelatedItem].[RelatedItemMain[.[Type] == "Up-sell")
```

The following expression verifies whether a row exists in a multirow attribute group:

```
exists(isNull([Item].[Ingredients].[Ingredient Name]) == false)
```

Example rules:

The following rule verifies that if the item attribute Pack Type is specified, then the GTIN attribute GTIN cannot be null.

```
If Expression:
isNull([Item].[Main].[PACK TYPE]) == false
```
Validation Condition:
exists(isNull([GTIN].[GTINMain].[GTIN])) == false

The following rule verifies that if the value of the attribute \texttt{TM} is \texttt{US}, then the GTIN relationship must be with a Customer named USFDA.

If Expression:
\[ [\text{Item}].[\text{Group1}].[\text{TM}] == \text{"US"} \]
Validation Condition:
\[ [\text{GTIN}].[\text{GTIN Main}].[\text{Party Type}] == \text{"Customer" AND} \]
\[ [\text{GTIN}].[\text{GTIN Main}].[\text{Party Name}] == \text{"USFDA"} \]

to\_item\_class

Syntax

\texttt{to\_item\_class("item class name")}

Invoked when the item class of an item is being changed. In a validation expression, returns TRUE if the name of the new item class matches \texttt{"item class name"}, which must be enclosed in quotation marks.

Example:

The following example returns TRUE if the new item class is \texttt{TCCHLDVR}

\texttt{TO\_ITEM\_CLASS("TCCHLDVR")}

from\_item\_class

Syntax

\texttt{from\_item\_class ("item class name")}

Invoked when the item class of an item is being changed. In a validation expression, returns TRUE if the name of the original item class matches \texttt{"item class name"}, which must be enclosed in quotation marks.

Example:

The following example returns TRUE if the original item class is \texttt{TCParent}:

\texttt{FROM\_ITEM\_CLASS("TCParent")}

Combined example:

The following example prevents changing the item class of an item from \texttt{Pneumatic\_Pumps} to \texttt{Hydraulic\_Pumps}.

\textbf{Severity: Reject}

If Expression:
\texttt{FROM\_ITEM\_CLASS("Pneumatic\_Pumps") && TO\_ITEM\_CLASS("Hydraulic\_Pumps")}

Descriptive Flexfields

Descriptive flexfields do not belong to any attribute group, and are accessed using the \texttt{FlexField} segment code.

\[ [\text{<Entity Name>}).\text{Flexfield}<\text{Flexfield segment code}>] \]

Example:

\[ [\text{ChangeHeader}].\text{Flexfield}[\text{Product\_Line}] \]
Comparison Operators
The comparison operators available in item rules are listed below.

- `==` (equals)
- `!=` (not equals)
- `<` (less than)
- `<=` (less than or equal)
- `>` (greater than)
- `>=` (greater than)

Comparison operators compare two Boolean expressions and return TRUE or FALSE, depending on the result of the comparison. If one or both expressions are null, then the comparison returns null. To check for null values, use the comparison function `isnull`.

String comparison is case-insensitive. For case-sensitive comparison use the string function `compare`.

Syntax:

```plaintext
expression1 == expression2  (equals
expression1 != expression2
expression1 < expression2
expression1 <= expression2
expression1 > expression2
expression1 >= expression2
```

`isnull`

Syntax:

```plaintext
isnull(expression)
```

Returns TRUE if its argument is null, returns FALSE otherwise. This function lets you explicitly test whether a value is null. Unlike other functions, it is not be ignored if the value of the argument is null.

The logical function `isnull()` implements the following truth table for `expression`.

<table>
<thead>
<tr>
<th>expression</th>
<th>isnull(expression)</th>
</tr>
</thead>
<tbody>
<tr>
<td>null</td>
<td>T</td>
</tr>
<tr>
<td>not null</td>
<td>F</td>
</tr>
</tbody>
</table>

Item Rule Utility Functions
Use these functions in building more complex expressions.

`auto_sequence`

Syntax:
auto_sequence("Sequence Name", starting_num, increment_by)

Searches for *Sequence Name* in the database tables. Enclose the sequence name in quotation marks to preserve space characters in the name. The sequence name is not case-sensitive. If the sequence exists, then the function returns the next value from the sequence. If the sequence does not exist, then the function creates a sequence with the specified name and returns *starting_num*. Further numbers in the sequence are incremented by *increment_by*. This function can be used in defining the return value of assignment and validation rules.

Example: The following expression used as the return value of a rule whose target attribute is [General attributes].[EDC Number] produces a sequence of numbers beginning at 1000 and increasing by 5 for each new item:

```
auto_sequence("EDC Number", 1000, 5)
```

**between**

Syntax:

```
between(value, minimum, maximum)
```

Returns TRUE if *value* falls between *minimum* and *maximum*. Returns FALSE otherwise. All arguments are of type Number, Date, or DateTime.

**decode**

Syntax:

```
decode(expression, search1, result1, [search2, result2, ...], [default])
```

Compared *expression* to the specified series of *search* arguments, one at a time, and returns the corresponding *result* when the first match is found. If no match is found, returns *default*. If *default* is omitted, returns null. Requires specification of at least *expression*, *search1*, and *result1*. You can specify an unlimited number of pairs of *search* and *result* arguments. The *default* argument is optional.

Example:

The following example returns RED COLOR if the item's body color is RED, BLUE COLOR if it's BLUE, and causes a rejection if none of those colors is matched.

```
Severity: Reject
If Expression: decode([Item].[BODY ATTR].[COLOR], "RED", "RED COLOR", "BLUE", "BLUE COLOR", "NONE") == "NONE"
```

**getCustomObjectValue**

Syntax:

```
getCustomObjectValue(
custObjName,  
custObjReturnAttrName,  
custObjQueryAttrName1, Value1,...
custObjQueryAttrNameN, ValueN)
```

Fetches a value from attributes of a custom object *custObjName* that is defined using Application Composer. The value is fetched from the field named *custObjReturnAttrName* of the custom object. The custom object instance to fetch the value from is identified by matching *Value1* in the field named *custObjQueryAttrName1*, using the == operator. *Value1* can be a literal value, or can refer to an attribute, such as [ITEM].[AG1].[A1]. You can specify additional matching criteria by providing matches for *custObjQueryAttrNameN* against *ValueN*, and so on.
in
Syntax:
\[
\text{in}(\text{expression}, \text{value1}, \text{value2}, \ldots)
\]
Returns TRUE if \text{expression} is found in \text{value1}, \text{value2}, or other following value arguments. Returns FALSE otherwise. Requires specifying at least \text{expression} and \text{value1}. You can specify an unlimited number of \text{value} arguments.

Example:
The following example returns TRUE if “RED” is contained in the value of either of the specified attributes for colors.

\[
\text{If Expression: in("RED", [Item].[BODY ATTR].[COLOR], [Item].[COVER ATTR].[COLOR])}
\]

Note: The functions \text{in() and not_in()} accept arguments of either String or Number. All arguments must be of the same type. The type of the first argument determines the type assigned to the subsequent arguments.

not_in
Syntax:
\[
\text{not_in}(\text{expression}, \text{value1}, \text{value2}, \ldots)
\]
Returns TRUE if \text{expression} is not found in \text{value1}, \text{value2}, or other following value arguments. Returns FALSE otherwise. Requires specifying at least \text{expression} and \text{value1}. You can specify an unlimited number of \text{value} arguments.

Example:
The following example returns TRUE if “RED” is not contained in the value of either of the specified attributes for colors.

\[
\text{If Expression: not_in("RED", [Item].[BODY ATTR].[COLOR], [Item].[COVER ATTR].[COLOR])}
\]

to_number
Syntax:
\[
\text{to_number}(<\text{string})
\]
Returns \text{string} as a value of type Number.

Example:
The following example returns the count of an item as a string.

\[
\text{to_number}([\text{Item}.[PHYSICAL PROP].[COUNT])}
\]

Date Operators
The + (plus) and - (minus) operators add or subtract the specified number of days from a date. A single number is interpreted as a number of days. You can also use dates in the format specified by the standard ISO 8601.

Syntax:
\[
\text{expression1 + expression2}
\]
\[
\text{expression1 - expression2}
\]

Example:
The following example expresses a time 3 days after the Item Lead Time

\[ \text{Item}.\text{Logistics}.\text{LeadTime} + 3 \]

### Using Custom Object Data in Rules: Explained

You can use the function `getCustomObjectValue()` to access data stored in a custom object created in the Oracle Fusion CRM Application Composer.

Consider the following example use case for custom objects. Assume that your company uses a catalog for export-controlled items. You need to assign your inventory items to the catalog based on complex rules that take into account both item attributes and non-item attributes, such as trade agreements. You maintain the item attributes as operational attributes and extensible flexfields in the Product Information Management work area. You maintain the trade agreement attributes in custom business objects designed using Application Composer.

Item rules can refer to both the item attributes and the non-item attributes, to determine if an item should be assigned to the export-controlled items catalog. For accessing the attributes maintained in custom objects, item rules use the function `getCustomObjectValue()`.

In Application Composer, you have defined a custom object called Selling Restrictions, containing a matrix of selling restrictions by target market. You want to refer to that data to determine whether or not an item can be sold, based on its target market. Assume that the example custom object Selling Restrictions contains the data in the following table.

<table>
<thead>
<tr>
<th>Target Market</th>
<th>Restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>EU</td>
<td>No</td>
</tr>
<tr>
<td>North America</td>
<td>Yes</td>
</tr>
<tr>
<td>APAC</td>
<td>Yes</td>
</tr>
<tr>
<td>Middle East</td>
<td>No</td>
</tr>
</tbody>
</table>

The following example fetches the value of Restriction from Selling Restrictions where the Target Market is North America. That fetched value for Restriction is Yes.

```java
getCustomObjectValue(
    "Selling Restrictions",
    "Restriction",
    "Target Market", [ITEM].[Market Attributes].[Target Market])
```

Your item rules can use complex business rules to determine attributes for assignment or validation purposes. You can shift some of the data-induced complexity for business rules, such as deriving the restriction for a target market, or deriving the color family name for a given color shade, to custom business objects and then refer to those values in your business rules. This practice of decoupling business rules from the underlying data protects your business rules from changes in data sets, such as color shades being added or dropped, and simplifies maintenance of your business rules.

Another use of custom objects with item rules is to use data available in legacy systems that may not have built-in web services. Relevant data from such systems can be extracted and maintained in custom objects and then be referred to in your item rules. This practice greatly enhances the scope of data used in your business rules.
Related Topics

- Creating an Object: Example

Blending Rules

Blending Rules: Overview

To control which item attribute value is imported into production from multiple suppliers of the item, you can define blending rules, which are applied during import, and which determine which supplier’s attribute value to import, based on the blending priority that you define in the blending rules.

To use blending rules:

- Understand how blending rules affect import
- Examine an example of how blending rules affect import
- Define blending rule sets and rules, and enable the blending rule sets

How Blending Rules Affect Import: Explained

When you import the same item from multiple suppliers, specific attributes of the item might have different values, depending on the supplier. To control which attribute value is imported into production, you can define blending rules, which are applied during import, and which determine which supplier’s attribute value to import, based on the blending priority that you define in the blending rules.

During the import process, blending rules use the spoke system item relationships on a production item to identify the supplier product records to be blended with imported data. Blending rules are applied at the item level, in a specified order of preference among the spoke systems. If data quality is enabled, it runs to standardize the values of blended product records. When the import is uploaded to production, blended values overwrite the item attribute values in production.

Blending rules are not applied to product data for new items, since there is no existing data to be blended with new data. The spoke system item relationship used by blending rules to relate a supplier item to a production item is not created until an item is imported into production.

Blending rules are applied in the following business events:

- When an existing supplier provides updates to product data that was imported earlier.
- When a new supplier provides data for an existing item.

Blending rules operate during import if:

- Existing spoke system cross-references are found in the production database.
- New spoke system cross-references are established as a result of matching with a production item containing spoke system cross-references with other spoke systems.

You can choose to apply blending rules to attributes in the following ways:

- All attributes in one or more attribute groups. (This is the most common case.)
- All attributes associated with one or more item classes.
• One or more attributes from a single attribute group.

After blending is completed, the blended item record overwrites production data for the item.

General principles guiding the application of blending rules include:

• Blending rules run only if any of the suppliers mentioned in the blending rule has provided data into the supplier stage. If the higher priority supplier has not provided any data, then whatever is provided by the lower priority or other suppliers (suppliers not mentioned in the blending rules) will be imported.

• If a blending rule is written on an attribute then that rule will run only if that attribute is part of import. The attribute can be part of import because values are provided for that attribute in import.

• If a blending rule is written on an attribute group then that rule will fire only if any of the attributes of that attribute group are part of import. The attributes can be part of import because values are provided for those attributes.

• If a blending rule is written on an item class then that rule will run only if items of that item class or its child item classes are being imported.

• No updates will occur to the items staged in Oracle Fusion Product Hub Portal. Blending happens only within the import batch, and is then imported to production.

• If more than one rule exists on the same attribute, then the first rule in the master blending rule set will be run.

• Products in statuses Rejected or Draft do not get into the batch, so they do not have cross-references, and so they are not considered for blending.

• Blending rules defined for an item class which is at a higher level of the item class hierarchy will be inherited to child item classes.

Restrictions and validations on blending rules include:

• Blending rules apply only for the item entity, not for the item revision or supplier entities.

• Blending rules may be set up on any attributes of an item.

• Blending rule sets cannot be composite rule sets.

• Blending rules defined at the higher levels of an item class hierarchy are inherited by child item classes.

• Product Hub spoke and non-vendor spoke systems are not available as source systems for use in blending rules.

• Rule set impact analysis is not available for blending rule sets.

Related Topics

• Supplier Product Uploads: Explained

• Item Import Using Import Maps: Explained

• Item Batches: Explained

Blending Rule Sets and Rules: Procedures

Blending rules must be created as part of a blending rule set, and the blending rule set must be associated with the entity containing the attribute whose value you want to control through blending rules.

Defining Blending Rule Sets and Rules: Procedure

To define a rule set and rules for a blending rule:

1. In the Product Information Management work area, select the Manage Item Rule Sets task.
2. Select Create from the Manage Rule Sets task bar.
3. In the Create Rule Set dialog box, make the following required selections:
   - Set Composite to No.
   - Set Type to Blending.
4. When you set the Type to Blending, the available rule association types are restricted to attribute groups or item classes. In the Association Type field, select either Attribute Group or Item Class.
5. For an association type, you can select multiple attribute groups or item classes to associate with the rule set. In the Associations field, click the edit control.
6. In the Select Associations dialog box, click Add, to add an association with the rule set.
   - For an attribute group association, this entity is the attribute group containing the attribute.
   - For an item class association, this entity is the item class associated with the attribute group containing the attribute.
7. If you select Draft, then the rule set will not be triggered at all during import.
8. Click Save and Continue.
9. In the Rules tab of the Edit Rule Set page for the new rule set, select Create from the Actions menu.
10. In the Create Rule dialog box, enter an integer in the Sequence field. The sequence number determines the order in which the rule will be triggered when the rule set is triggered.
11. Using the Attribute list, select the attribute whose values you want to blend during import.
   - If you selected multiple associations, then the rule applies to all attributes in the attribute group or item class, and you can’t select an individual attribute. Accordingly the Attribute Group and Attribute lists are restricted to the value All.

   **Note:** The following rule fields are not used by blending rules:
   - Return Type
   - Target Business Entity
   - Target Attribute Group
   - Target Attribute

12. In the Details region for the rule, select Create from the Actions menu. Create a blending priority for the rule’s associated attribute by selecting a value from the Spoke System list and entering a number in the Priority field.
   - Lower priority numbers produce a higher blending priority. A spoke system with priority number 10 has a higher blending priority than a spoke system with priority number 20.
13. Repeat the previous step for each spoke system that supplies attribute values that you want to blend.
14. Define more rules in the rule set, as needed.
15. Click Save to save the rule set.
16. Add the rule set to the master blending rule set, to enable its operation during import.

**Enabling Blending Rule Sets: Procedure**
To enable the triggering of a blending rule, you must add its rule set to the master blending rule set, which is named BLENDBING_MASTER_RULE_SET. This composite rule set is predefined. You can’t create any other composite blending rule sets. Only rule sets with a Type of Blending can be added to this master set. When a blending rule set is added to the master blending rule set, its blending rules are triggered when you import items into Oracle Fusion Product Hub.
To add a rule set to the master blending rule set:

1. In the Product Information Management work area, select the Manage Item Rule Sets task.
2. On the Manage Rule Sets page, search for and select the rule set named BLENDING_MASTER_RULE_SET.
3. Select the rule set BLENDING_MASTER_RULE_SET.
4. Select Add from the Actions menu.
5. In the Select and Add: Rule Sets window, use the desired criteria to search for the rule sets that you want to add to the master.

   **Tip:** You can find all blending rule sets by specifying Blending in the Type field. You can search for your blending rule sets by specifying the association type or association name that you used in your rule sets.

6. Select the rules sets you want to add, then click OK to add them to the master blending rule set.
7. The rules sets you added are place at end of the sequence of blending rule sets in the master blending rule set, and are run in that sequence. If there are multiple blending rules defined for an attribute, only the first rule is run.
8. To change the place of a blending rule set in the sequence, select it and select Edit Sequence from the Actions menu. In the Edit Sequence dialog box, enter the desired sequence number in the Sequence field and click OK.

You must first edit the sequence of other rule sets to ensure that the desired sequence number is not already in use.

**Related Topics**
- Supplier Product Uploads: Explained
- Item Import Using Import Maps: Explained
- Item Batches: Explained

**Blending Rules: Example**

The following example demonstrates the effect of blending rules.

**Scenario**

For this example, assume that you have defined the blending rules shown in the following list.

- **Blending_Rule_1**
  - Item Class: Televisions
  - Attribute Group: Specs
  - Attribute: Frame Description
  - Priority -- Spoke System:
    - 1-- Vendor1 Spoke
    - 2-- Vendor2 Spoke

- **Blending_Rule_2**
  - Item Class: Televisions
  - Attribute Group: Specs
  - Attribute: Resolution
• Priority -- Spoke System:
  • 1 -- Vendor2 Spoke
  • 2 -- Vendor1 Spoke

• Blending_Rule_3
  • Item Class: Televisions
  • Attribute Group: Specs
  • Attribute: Weight
  • Priority -- Spoke System:
    • 1 -- Vendor3 Spoke
    • 2 -- Vendor4 Spoke

At the beginning of the import process, Vendor1 loads data in Product Hub Portal for the attributes **Frame Description** and **Resolution**, as shown in the following list.

• Vendor1
  • Spoke Item:
    • Vendor1 TV Item
    • Attribute: Frame Description
    • Value: Plastic body with back hinge
    • Attribute: Resolution
    • Value: 1930x1024

You import the loaded data. Since Vendor2 has not loaded any data in Product Hub Portal, the values loaded by Vendor1 are imported for both of the attributes, Frame Description and Resolution. Consequently, the attribute values in production data are as follows.

• Production data in Product Hub:
  • Production item in Product Hub:
    • TV_1
    • Spoke Item cross-reference
      • Vendor1 TV Item
      • Attribute: Frame Description
      • Value: **Plastic body with back hinge** (initial value, from Vendor1)
      • Attribute: Resolution
      • Value: **1930x1024** (initial value, from Vendor1)
At a later time, Vendor2 loads data in Product Hub Portal for the attributes Frame Description and Resolution, as shown in the following list.

- **Vendor2**
  - Spoke Item:
    - **Vendor2 Television**
      - Attribute: Frame Description
        - Value: Plastic with hinge
      - Attribute: Resolution
        - Value: 2930x1034

Then you try to import the data loaded by Vendor2. You determine that there is a match with the existing production item TV_1, so you create a new spoke system cross-reference with TV_1. Since Blending_Rule_1 gives blending priority for the attribute Frame Description to Vendor1, Vendor1’s value for that attribute remains as previously imported into production. Since Blending_Rule_2 gives blending priority for the attribute Resolution to Vendor2, Vendor2’s value for that attribute is imported into production. The resulting data imported into production, as shown in the following list.

- **Production data in Product Hub:**
  - Production item in Product Hub:
    - **TV_1**
      - Spoke Item cross-references
        - **Vendor1 TV Item**
        - **Vendor2 Television**
      - Attribute: Frame Description (Vendor1 has blending priority)
        - Value: **Plastic body with back hinge** (from Vendor1, unchanged)
      - Attribute: Resolution (Vendor2 has blending priority)
        - Value: **2930x1034** (from Vendor2)

At a later time, both Vendor1 and Vendor2 update their data in Product Hub Portal, including changed values for Frame Description and the addition of the new attributes Height and Weight, as shown in the following lists.

- **Vendor1**
  - Spoke Item:
    - **Vendor1 TV Item**
      - Attribute: Frame Description
        - Value: **Plastic body of PVC make with metallic back hinge** (updated)
      - Attribute: Resolution
        - Value: 1930x1024
      - Attribute: **Height** (new)
        - Value: **2 Feet** (new)
Then you import the updated data from Vendor2 (but not from Vendor1). The resulting data is imported into production, as shown in the following list.

- Production data in Product Hub:
  - Production item in Product Hub:
    - TV_1
  - Spoke Item cross-references:
    - Vendor1 TV Item
    - Vendor2 Television
    - Attribute: Frame Description (Vendor1 has blending priority)
      - Value: Plastic body with back hinge (original import, from Vendor1)
    - Attribute: Resolution (Vendor2 has blending priority)
      - Value: 2930x1034 (unchanged, from Vendor2)
    - Attribute: Weight (new, from Vendor2)
      - Value: 5 kg (new, from Vendor2)

Note the following about the import into production data:

- Since Blending_Rule_1 gives blending priority for the attribute Frame Description to Vendor1, Vendor1’s value for that attribute is used instead of Vendor2’s updated value, which is not imported.
- Because Vendor1’s updated data was not imported, Vendor1’s value for the attribute Frame Description remains as originally imported into production.
- The new attribute Weight is imported from Vendor2. Even though Blending_Rule_3 is defined on that attribute, the rule is not triggered because neither Vendor3 or Vendor4 uploaded data in Product Hub Portal, and no spoke system cross-references for these vendors exist or are created by matching during import.

Related Topics

- Supplier Product Uploads: Explained
- Item Import Using Import Maps: Explained
- Item Batches: Explained
7 Define New Item Requests

New Item Requests: Overview

Before you can use new item requests in Oracle Fusion Product Hub, you must complete these tasks in the Setup and Maintenance work area:

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Change Order and New Item Request Header Descriptive Flexfields</td>
<td>Used to manage the header-level descriptive flexfields for change orders and new item requests.</td>
</tr>
<tr>
<td>Manage New Item Request Type Details</td>
<td>All new item requests are assigned a new item request type. You must define at least one new item request type to use new item requests.</td>
</tr>
<tr>
<td>Manage Change Order and New Item Request Line Descriptive Flexfields</td>
<td>Used to manage the line-level descriptive flexfields for change orders and new item requests.</td>
</tr>
<tr>
<td>Manage Task Configurations for Supply Chain Management</td>
<td>Define the task configurations for the new item request approval workflow.</td>
</tr>
<tr>
<td>Manage Approval Groups for Supply Chain</td>
<td>Define approval groups for the new item request approval workflow.</td>
</tr>
</tbody>
</table>

New Item Requests Workflow Statuses: Explained

There are five predefined New Item Request (NIR) workflow statuses which enable you to perform various tasks when a new item request is created. You cannot create new statuses or delete statuses.

The four statuses are:

- Open
- Definition
- Approval
- Scheduled
- Completed

Open

New item request attributes and items are defined and updated when the workflow status is open. A seeded request comment notification is associated with this status and the new item request cannot move to the next status until this notification has been responded to. The notification will be sent to the assignee and the requester of the new item request.
However, to bypass the notification, an administrator can select to skip this notification on the new item request type. Assignees would then need to promote the new item request to the next status manually or on the new item request type.

Definition

In the definition workflow status, you define the item information, such as specifications, structures, packs and so on. Additional items cannot be added in this workflow status. The definition workflow steps are defined at the item class level. Based on the definition steps, a notification is sent to those responsible for defining the item. Once the item is defined, you then promote the new item request to approval status manually or to be automatically approved in the BPEL process.

Approval

When approval is granted, the new item request cannot be modified. Approval notifications are sent to the approvers based on the rules set up in Approval Management Extensions (AMX), if the assignment method is rule-based, or to a pre-defined set of approvers, if the assignment method is user-defined. After all the required approval is received and the status is Scheduled, a job is triggered and the status is automatically changed to Completed.

Scheduled Status

After approval, the NIR is automatically promoted to a Scheduled status. Automatic promotion and demotion is set up on new item request type, otherwise the promotion is manual.

Completed

After all the required approvals are received and the status is Scheduled, a job is triggered and the status is automatically changed to Completed. It cannot be modified.

New Item Request Definition Phase: Explained

During the definition phase, definition notifications are routed to participants for them to complete the item’s definition tasks.

Define an Item

From the Manage New Item Requests page, search for the new item request.

Launch the new item request from the Search Results table by clicking on a new item request link. That will take you to the New Item Request page.

Click on the Details sub tab and select the definition row. In the definition Workflow Details table, select Expand All from the Actions menu.

Click the Go to Task icon for the entity that needs definition. The item page appears where you enter the item information. The item definition page is rendered dynamically for each participant, where you can enter only the attributes or item entities for which you are responsible.

Note: You can also navigate and drill into an item detail page directly from a new item request definition notification. A read-only item page will open a separate browser window.
You can identify various item details as mandatory at each step. This will ensure that item information required for a downstream step is defined and available for use.

Any item definition that is available to be set up for definition through new item request, can be identified as mandatory while completing the definition steps at the item class, including:

- Operational
- Extensible flexfield attributes
- Structures
- Packs
- Supplier associations
- Organization assignments
- Attachments
- Catalog category assignments
- Relationships

New Item Request Approval Phase: Explained

During the approval phase of a new item request, approval notifications are routed to the participants.

You can set up the new item request type so that a request only needs to be approved by a single member of a user group. On the Manage New Item Request Type Details page of the Setup and Maintenance work area, select the Approval step on the Workflow tab. Then for an approval activity in the step’s status details, set Response Required From to One. When one member of a group approves the request, the notifications to other approvers in the group are withdrawn.

Approval notifications can be addressed to individual users or to multiple users in a predefined user group. From the notification, users can approve or reject the new item request.

For each new item request header, you can select an assignment method in the Approval status details for the Header approval, as either rules-based or user-defined. If you choose user-defined assignment, you then select an approver by using the Assigned To control.

You can also assign approvers as optional approvers. A single optional approver can reject a change order, but approvals from optional approvers are ignored. To assign an optional approver, select an approver by using the Assigned To control on the Optional approval row of the Approval status details.

When an approval is granted, the new item request cannot be modified. After the new item request is approved, the status is changed to Scheduled, through a scheduled process, and when its effective date is reached, its status is changed to Completed, also through a scheduled process.

New item approval rules are defined and managed through the Approval Management Engine. For information on how approval rules are defined and managed, refer to the Oracle Fusion Middleware Developer’s Guide for Oracle SOA Suite.

**Note:** The assignment method needs to be rule-based in order for approval rules to be utilized.

**Related Topics**
- New Item Requests: Explained
- New Item Request Notifications: Explained
Defining Entry and Exit Criteria for New Item Requests: Procedure

You can define criteria that govern when a new item request can exit the current workflow status or enter into the next status. You define such entry and exit criteria in product rules, and then select those criteria rules when defining the new item request type.

Defining entry and exit criteria for a new item request type requires the following tasks:

1. Defining the criteria in a validation rule set.
2. Assigning the criteria to the new item request type’s workflow.

Defining the Criteria in a Validation Rule Set

Define one or more rule sets that validate your criteria for new item request status entry and exit. Criteria can be based on attributes or on descriptive flexfields.

You define the specific criteria for validating the criteria by creating product rules. You can write validation expressions in the product rules using references to the following business entities:

<table>
<thead>
<tr>
<th>Business Entity</th>
<th>Provides Access To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Header</td>
<td>Descriptive flexfield segments on new item request header rows. The attribute group Change Header Main, which contains attributes corresponding to the basic attributes of a new item request header: Priority, Reason, Need-by Date, Requested By, and Description</td>
</tr>
<tr>
<td>Change Line</td>
<td>Descriptive flexfield segments (but not attributes) on new item request lines</td>
</tr>
</tbody>
</table>

Example: Define a product rule that allows the change order workflow to exit from Open status only if its Reason is Cost and its Priority is High.

1. From the Items work area, select the Manage Item Rule Sets task.
2. Create a rule set. In the Create Rule Set dialog, you must make the following selections.
   - For Type, select Validations.
   - For Association Type, select New Item Request.
   - For Association Name, select New Item Request.
3. On the Edit Rule Set page, create a rule. The Severity is automatically set to Reject.
4. In the If Expression field, right-click and select Insert Attribute. Using the Insert Attribute dialog, select the Business Entity, Attribute Group, and Attribute that insert the following expression:

   \([\text{ChangeHeader}].[\text{ChangeHeaderMain}].[\text{ReasonCode}]\)

5. Add the rule code \(= \text{"Cost"}\), to form the following complete comparison expression:
6. In the Validation Condition field, use the same procedure to insert the following expression:

\[[\text{ChangeHeader}].[\text{ChangeHeaderMain}].[\text{ReasonCode}] == \text{"Cost"}\]

7. Optionally, add a user message, such as the following:

New item requests related to cost must have high priority to be considered for approval.

8. Validate the rule’s syntax, by clicking the Validate button.

Assigning the Criteria to the New Item Request Workflow

Modify the new item request type to use a validation criteria rule set.

After defining a rule set that validates your new item request status criteria, you can assign that rule set as the entry or exit criteria for the new item request type statuses related to the approval workflow. The new item request type must be already associated with the rule set in order to use the rule set in the new item request type.

You can assign criteria to a new item request’s workflow statuses, beginning with its exit from Open status through its entry to Approval status. The status types that can have entry or exit criteria are listed in the following table:

<table>
<thead>
<tr>
<th>Status Type</th>
<th>Entry Criteria?</th>
<th>Exit Criteria?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Definition</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Approval</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Example: Select the entry criteria for the workflow status Approval.

1. From the Setup and Maintenance work area, search for and select the Manage New Item Request Type Details task, which enables you to edit the new item request type. There is only a single new item request type.

2. On the Workflow tab of the Manage New Item Request Type Details page, select the status Approval.

3. The choice list in the Entry Criteria column does not yet have a selected value. Select the name of the rule set that you created, which defines the entry criteria for this status of the workflow for new item requests.

Now, when a new item request of this type is submitted for the Approval status, it will be rejected by the validation criteria rule set if its Priority value is not High.

Related Topics

- New Item Requests: Explained
- Defining Entry and Exit Criteria for Change Orders: Procedure
- Item Rule Syntax: Explained
8 Define Product Spoke Systems

Product Spoke Systems: Overview

Before you can use Product Spoke Systems in Oracle Fusion Product Hub, you must complete this task in the Setup and Maintenance work area:

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define Source Systems for Product Management</td>
<td>Spoke systems enable users to identify the source of the data they are imported into the Oracle Fusion database or published from the Oracle Fusion database. The terms spoke system and source system are used interchangeably.</td>
</tr>
</tbody>
</table>

Product Spoke Systems: Explained

Spoke systems (also called Source Systems) enable users to identify the source of the data they are imported into the Oracle Fusion database or published from the Oracle Fusion database.

Spoke System Setup

The task to set up spoke systems is the Manage Trading Community Source System task in the Set up and Maintenance work area.

A spoke system is identified by a unique code and a name. A spoke system has description and type as well. When the spoke system is defined, the types of entities that will be supported are selected from the following list:

- Items
- Trading Community Members
- Order Orchestration and Planning
- Assets

The spoke system used by Oracle Product Hub must have the Items entity enabled.

Item Publication Criteria:

- **Publish Only Changed Entities**: only the changed item entities will be published.
- **Revisions**: controls what revisions will be published for the items; choices are: Current Revision or Current and Future Revisions
- **Days in the Future**: number of days in future that will be used to determine what future revisions will be published.
- **Item Entities**: select the child entities of the items that will be published for the items, by default on the item attributes will be published. The entities that are published:
  - Attributes
  - Attachment URL
Supplier Site Associations
Item Category Assignments
Item Relationships
Structures
Packs
Item Revisions

**Item Select Rules:** one or more rules that are based on business entities that include:

- **Item Class Business Entity Rule:** equal/does not equal to an Item Class with option to select to include the child item classes for the item class in the rule
- **Catalog Business Entity Rule:** equal/does not equal to an Item Class with option to select to include the child item classes for the item class in the rule
- **Organization Business Entity Rule:** equal/does not equal to an Organization
- **Organization Hierarchy Business Entity Rule:** equal/does not equal Organization hierarchy and sub-organization within the hierarchy

**Item Validation Rules:** a validation rule set that will be used to determine which items are published. The rule set may contain one or more rules and is defined using the Manage Item Rule Sets task.

- Example: A validation that rule that would publish only the items with an active status: **If Expression:**
  
  $\text{INVENTORY\_ITEM\_STATUS\_CODE} == 'Active'$

- Item Validation Rules are evaluated prior to publishing.
9 Define Advanced Catalogs

Advanced Catalogs: Overview

Before you can map catalogs in Oracle Fusion Product Hub, you must complete this task in the Setup and Maintenance work area:

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Catalog Mappings</td>
<td>The Manage Catalog Mappings task is used to create a mapping between one catalog and another catalog or between a catalog and the root item class. These catalog mappings contain category mappings between the categories of one catalog to the categories of another catalog or between categories of a catalog and item classes under the root item class.</td>
</tr>
</tbody>
</table>

Catalog Mappings: Explained

The Manage Catalog Mappings task is used to create a mapping between one catalog and another catalog or between a catalog and the root item class. These catalog mappings contain category mappings between the categories of one catalog to the categories of another catalog or between categories of a catalog and item classes under the root item class.

While creating a catalog mapping, users select a source catalog and a target catalog. The combination of the source catalog and target catalog should be unique, meaning, users cannot create two mappings with the same source and target catalogs. Once the source and target catalogs are mapped and the mapping is created, users can create mappings between the categories of the source catalog and the categories of the target catalog within this catalog mapping. For the category mappings, only one mapping can be created for a particular source category, meaning, users cannot create two mappings with the same source category. However, different source categories can be mapped to the same target category.

Item Class to Category Mapping

You can use GPC (global product classification) or any other category classification to model a catalog category hierarchy in Product Hub. The catalog representing this hierarchy can then be used for creating the mapping between the catalog categories and item classes. This mapping can then be used to derive the correct item class, if the category is specified. The item class can then be used to import data into Product Hub and to display proper attributes in Product Hub Portal.

The flow follows:

1. Product data stewards create a catalog hierarchy depicting the desired classification using the Create Catalog task in the Product Information Management work area.
2. Next, update the item profile option, Default Catalog for Item Import, in the Setup and Maintenance work area to set the newly defined catalog as the catalog that should be used while importing data that has Catalog specified instead of item class for the mapping.
3. A mapping between this catalog and the item class hierarchy is then made using the Manage Catalog Mappings task in the Setup and Maintenance work area. The catalog to item class mapping must specify the source catalog as the newly created catalog and the target catalog as an item class hierarchy. The category mappings created within this catalog to item class mapping must specify the source category as one of the categories from the new catalog and the target category as the item class.
4. In Product Hub Portal, while uploading or creating products, supplier users select a category for the products. The Create and Edit Product pages in Product Hub Portal display the Category field instead of the Item Class field. The drop-down list will list all the categories of the source catalog irrespective of which catalog categories are mapped in the catalog to item class mapping setup. The application will derive the correct item class from the selected category by referring to the category mappings created in step 3.

5. The Create and Edit Product pages use the derived item class to display the appropriate product attributes to the suppliers. The attributes are derived from the values selected on the Product Hub Portal tab in the Create Item Class task in the Setup and Maintenance work area.
10 Define Item Mass Update Configuration

Item Mass Update Configuration: Overview

Before you can use item mass updates in Oracle Fusion Product Hub, you must complete this task in the Setup and Maintenance work area.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Item Import Formats</td>
<td>Used to define import formats to be used when performing item mass updates.</td>
</tr>
<tr>
<td>Configuration</td>
<td></td>
</tr>
</tbody>
</table>

Item Import Formats Configuration: Explained

An import format identifies those main and user-defined attributes in an item class whose values are imported into the application using a spreadsheet. Consequently, when you import item data from a spreadsheet, the items are all imported into the particular item class for which the item format was defined. You can edit an import format after it is created.

While all the mandatory attributes will be automatically added to the import format, users can also pick and choose which other attributes to be included as part of the import format.

While setting up the import formats, you can selectively choose which item attributes to be included. All of these attributes defined in an import format will get added to the ADFdi spreadsheet when it is generated while adding the items to an item batch.

You can also optionally inherit import formats defined for the parent item classes while creating a new import format. This will inherit all the attributes of the parent item class import format to the import format being defined. This helps in maintaining the various import formats across the item class hierarchy.

Additionally you can mark an import format as inactive or active. This helps to selectively hide or display the import formats for downloading the ADFdi spreadsheet until the setup of the import format is complete.
11 Define Item Import Batch Configuration

Item Import Batch Configuration: Overview

Before you can create item import batches in Oracle Fusion Product Hub, you must complete this task in the Setup and Maintenance work area:

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Item Import Formats</td>
<td>Configuration: Used to define import maps to be used during the item import batch process.</td>
</tr>
</tbody>
</table>

Item Import Formats Configuration: Explained

An import format identifies those main and user-defined attributes in an item class whose values are imported into the application using a spreadsheet. Consequently, when you import item data from a spreadsheet, the items are all imported into the particular item class for which the item format was defined. You can edit an import format after it is created.

While all the mandatory attributes will be automatically added to the import format, users can also pick and choose which other attributes to be included as part of the import format.

While setting up the import formats, you can selectively choose which item attributes to be included. All of these attributes defined in an import format will get added to the ADFdi spreadsheet when it is generated while adding the items to an item batch.

You can also optionally inherit import formats defined for the parent item classes while creating a new import format. This will inherit all the attributes of the parent item class import format to the import format being defined. This helps in maintaining the various import formats across the item class hierarchy.

Additionally you can mark an import format as inactive or active. This helps to selectively hide or display the import formats for downloading the ADFdi spreadsheet until the setup of the import format is complete.
12 Define Audit History for Product Management

Audit History for Product Management: Overview

Before you can use audit policies in Oracle Fusion Product Hub, you must complete this task in the Setup and Maintenance work area.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Audit Policies</td>
<td>The Manage Audit Policy task lists Product Hub business objects and their attributes available for audit. You can re-calibrate your audit policy to suit changing business needs.</td>
</tr>
</tbody>
</table>

Audit Trail: Explained

Audit Trail is a functionality that allows users to identify a collection of Product Hub business objects (ex: Items) as business critical to be tracked for data changes (ex: marking the item as purchasable) and object definition changes (ex: adding an EFF). In other words, an audit policy is a collection of Product Hub business objects and the respective attributes that require continuous monitoring of changes to its data and definitions. Users can scrutinize change history on-line or off-line by exporting it into spreadsheet applications. In order to audit changes, an audit policy must be defined.

The Manage Audit Policy task lists all the Product Hub business objects and their attributes available for audit. You can define your audit policy by selecting the objects and the attributes of your change-tracking interest. For auditing descriptive flexfields, select the Additional Attributes check box.

Note that you can make adjustments to your audit policy as and when your business needs change.

When you make changes to Product Hub business objects listed in the audit policy, then the application logs who (user) changed what (add or update or delete) and when (date and time).

These are the data changes to business objects that cause a change to be logged:

- Data inserts
- Data updates
- Data deletions

The changes will be logged only for the objects and attributes that are members of the audit policy.

Note that Audit trail is not the same functionality as the New Item Request or Change Order functionality. While new item requests and change orders are approval oriented changes, Audit Trail involves the auto-logging of data change events that were triggered by any functionality, including new item requests and change orders.
The following table lists the search parameters used in the audit and the outcome of their selection in the detailed report.

<table>
<thead>
<tr>
<th>Search Parameter</th>
<th>Result of Selection</th>
</tr>
</thead>
</table>
| Business Object Type              | • Narrows the search results to that specific business object within the selected product.  
• Enables the Show Attribute Details check box to display.  |
| Include Child Objects             | Displays all of the child objects that were listed under the business object when the audit was set up.                                               |
|                                   | **Note**: This displays the objects at the immediate parent-child level only. To view the children at subsequent levels, select the child object as the business object type and search again. |
| Show Attribute Details            | • Displays the name of each attribute that users either created, updated, or deleted, and the corresponding old and replaced values.  
• Enables the attribute list so that users can select a specific attribute and view its change record.  
• Enables the Show Extended Object Identifier Columns check box to display. |
| Show Extended Object Identifier Columns | Displays the instances (contexts) in which the business object was used. The context values identify the objects and the transactions in which they were used. Each context is unique and assigns a unique description to the business object. For example, if an item is made purchasable in a child organization, then the child organization is listed as the context. |

You can access the Create Data Audit History Reports task to view the tracked changes on-line or export them as a .CSV file for off-line review using spreadsheet applications.
Chapter 13
Define Product Lifecycle Management

Define Product Innovation

Class Management in Oracle Innovation Management: Explained

Class Management is the definition of classes, class hierarchies, and class codes to establish reusable business objects. This topic introduces Class Management for Ideas, Requirements Specifications, Concepts, and Portfolios in Oracle Innovation Management.

The tasks addressed in this topic are:

- Manage Product Idea Classes
- Manage Product Requirement Classes
- Manage Product Concept Classes
- Manage Product Concept Component Classes
- Manage Product Portfolio Classes

Class

Use classes and subclasses to define business object types.

When you create a class, the class name that you provide is stored and used as an object type, at the time of business object creation.

Select a class to edit the class name and description. The Object Creation Allowed Indicator in the Edit Class page controls the possibility of creating business objects of the current class value. Select the indicator to ensure that the class name is available to use as a type when creating a business object.

Class Code

A Class Code is a constant and unique value associated with each class across Oracle Innovation Management and associated PLM systems.

You can define a class code only once, when creating a class, as it is used during integration with external systems, and is required to remain a consistent internal code.

Note: You cannot edit the class code after class creation. However, you can delete the existing class, if it was not already used to create an object, and create a class with the required class code.

Class Hierarchy

Class Hierarchy enables you to group and search for classes, based on class values or business objects types.

Select a class in the Manage Class page to view the class hierarchy in the Edit Class page.
Innovation Management Lookups: Explained

Oracle Innovation Management provides lookups that you can use to define values in Requirements, Concept, and Portfolio modules during implementation.

This topic addresses the following tasks:

- Manage Product Requirements and Ideation Lookups
- Manage Product Concept Lookups
- Manage Product Portfolio Lookups

Use standard lookups in Oracle Innovation Management to define values such as type, status, priority, scope, compliance, resource pool, metrics, lifecycle phases, and rank.

Application statuses are also standard lookups. You can execute the following tasks as lookup tasks:

- Manage Product Idea Status
- Manage Product Requirement Status
- Manage Product Concept Status
- Manage Proposal Status
- Manage Portfolio Status

The following table details the standard lookups available in Oracle Innovation Management.

Lookup types with customization level **System** do not allow you to add or delete lookup codes. However, you can edit the **Meaning** and **Description** fields of the existing lookup codes.

<table>
<thead>
<tr>
<th>Application</th>
<th>Module</th>
<th>Lookup Type</th>
<th>Lookup Code Meaning</th>
<th>Customization Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Requirements and Ideation</td>
<td>Ideas</td>
<td>Status</td>
<td>Pending, Accepted, Rejected, Implemented, In Progress, Review</td>
<td>User</td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Requirements and Ideation</td>
<td>Requirements</td>
<td>Associated Product</td>
<td>Common Services, Product Concept Design, Product Lifecycle</td>
<td>User</td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td></td>
<td>Portfolio Management, Product Requirements and Ideation</td>
<td></td>
</tr>
<tr>
<td>Product Requirements and Ideation</td>
<td>Requirements</td>
<td>Requirement Fulfillment</td>
<td>Yes, No</td>
<td>User</td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Requirements and Ideation</td>
<td>Requirements</td>
<td>Requirement Priority</td>
<td>Must Have, Nice to Have, Should Have</td>
<td>User</td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Define Product Lifecycle Management

<table>
<thead>
<tr>
<th>Application</th>
<th>Module</th>
<th>Lookup Type</th>
<th>Lookup Code Meaning</th>
<th>Customization Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Requirements and Ideation Management</td>
<td>Requirements</td>
<td>Requirement Scope</td>
<td>Yes, No</td>
<td>System</td>
</tr>
<tr>
<td>Product Requirements and Ideation Management</td>
<td>Requirements</td>
<td>Requirement Status</td>
<td>Pending, Submitted, Released</td>
<td>System</td>
</tr>
<tr>
<td>Product Requirements and Ideation Management</td>
<td>Requirements</td>
<td>Comment Status</td>
<td>Open, Closed</td>
<td>System</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Proposal Business Unit</td>
<td>Business Unit</td>
<td>User</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Concept Product Type</td>
<td>New Product, Technology Evaluation, Product Redesign</td>
<td>User</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Component Product Type</td>
<td>Documentation, Electrical, Mechanical, Software, Tooling</td>
<td>User</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Concept Status</td>
<td>Draft, Submitted Approved, Converted</td>
<td>System</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Cost Category</td>
<td>Development, Production</td>
<td>System</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Cost Status</td>
<td>Actual, Projected</td>
<td>System</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Proposal Cost Types</td>
<td>Fixed, Labor, Material, Variable</td>
<td>System</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Funding Request For</td>
<td>Concept, Feasibility, Product, Prototyping</td>
<td>User</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Market Strategy</td>
<td>Differentiation, Neutralization, Optimization, Others</td>
<td>System</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Primary Justification</td>
<td>Enter New Markets, Enter New Regions, Exit Market, Expand Market Share</td>
<td>User</td>
</tr>
</tbody>
</table>

**Related Topics**

- Lookups: Explained
- How can I edit lookups?
Manage Planning Periods: Explained

Product portfolio planning period is the time period during which the portfolio objects collect data for analysis. Before defining a planning period, create planning period units using the Manage Product Portfolio Planning Period task in the Setup and Maintenance work area.

Provide start date, number of units and specify the duration, which can be either monthly or quarterly, to create planning period time units. The planning period time units are created with default labels. You can add time units for a previously created planning period unit by providing the number of units before the first unit or by providing the number of units after the last unit.

Note: After you have created a planning period unit specifying a duration, you cannot change the duration. However you can change names of the units.

To create a planning period, provide the start date planning period unit, end date planning period unit, and select the planning period unit from the choice list.

Impact of planning period and planning period units:

- The columns in the Manage Resource Capacity table is dynamically created and displayed based on the number of planning period units defined in the planning period for the portfolio.
- The timelines in the Schedule, Resource, and Launch charts are decided based on the number of planning period units defined in the planning period for the portfolio.

Manage Product Portfolio Metrics: Explained

Oracle Innovation Management offers you a list of predefined metrics that you can use for measuring portfolio performance. You can also set the minimum and maximum threshold values for these metrics in accordance with business requirements.

Metrics can be defined in the Manage Product Portfolio Metrics task within the Setup and Maintenance work area. Metrics marked as Enabled are made available for selection when you add metrics using Actions > Define Metrics. Enabled metrics that are also marked as Default appear in the locations described in the following table.

<table>
<thead>
<tr>
<th>Default metrics for</th>
<th>Appear here</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product proposals</td>
<td>In the Proposal Metrics selection dialog that opens when you click the 123 icon during proposal creation.</td>
</tr>
<tr>
<td>Products</td>
<td>As column headers in the Elements table.</td>
</tr>
<tr>
<td>Portfolios</td>
<td>As column headers in the Scenarios table.</td>
</tr>
</tbody>
</table>
**Note:** A portfolio is a grouping of product lines, most often within company Business Units. All proposals related to product lines within a portfolio should ideally be evaluated by a common set of metrics so that the evaluations are consistent and objective. If you define consistent metrics for a portfolio and proposals within that portfolio, portfolio metrics can be rolled up from proposals as needed.

Metrics are of three types:

- **Derived** - Calculated using fixed formulas. (Examples: Return On Investment, Internal Rate of Return)
- **Derived and rolled up** - Calculated using fixed formulas and derived from certain values that you enter. (Examples: Net Present Value, Actual Cost, Projected Cost).
- **User entered** - Entered by the user in the user interface. (Examples: Impact, Alignment, Risk Numeric)

The following portfolio metrics can be configured using Data Composer:

- Resource Value Index
- Cost Value Index
- Expected Commercial Value
- Resources Productivity Index
- Cost Productivity Index
- Portfolio Strategic Fit

**Related Topics**

- Net Present Value: How It Is Calculated
- Internal Rate of Return: How It Is Calculated
- Break Even Time: How It Is Calculated
- Payback Period: How It Is Calculated
- Configure Innovation Management: Explained

### Manage Portfolio and Product Rule Sets: Explained

Define portfolio and product rule sets to associate multiple rules together, and assign them to portfolio classes. The following table lists and describes the portfolio and product rule sets.

<table>
<thead>
<tr>
<th>Type of Rule Set and Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation</td>
<td>Define validation conditions based on attribute values. predefined business rules Logical expression Validation condition User message</td>
</tr>
</tbody>
</table>
## Define Product Development

### Product Development Components: How They're Configured

This topic describes the hierarchy of tasks required to set up Oracle Fusion Product Development.

#### Item Organizations and Items

The Product Development offering includes two mandatory task bundles: **Item Organizations** and **Items**.

#### Change Orders

The **Change Orders** set of tasks are optional, and you do not have to configure them for Oracle Product Development to operate. However, if your installation requires routing change orders for items, manufacturer items, and assemblies, ensure that you configure this set properly. The **Change Orders Class** and **Change Order Types** must be created for your Product Development installation to have change orders available.

#### Product Development Configuration

The Product Development Configuration tasks provide additional refinement to your Product Development installation. This task bundle does not originate from Oracle Product Hub.

### Related Topics

- Change Orders: Overview
- Item Classes: Explained
- Defining Product Development: Overview
- Change Order Approval Process: Explained
Product Development Lookups: Explained

Use the Manage Product Development Lookups task in Setup and Maintenance to configure standard lookups (including category, status, and lifecycle phase) for Oracle Product Development.

Note: Lookup types with System customization level do not allow you to add or delete lookup codes. However, you can edit the Meaning and Description fields of their existing lookup codes.

This table lists and describes lookup types in Product Development work area.

<table>
<thead>
<tr>
<th>Application</th>
<th>Module</th>
<th>Meaning (Lookup)</th>
<th>Meaning (Codes)</th>
<th>Customization Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Development</td>
<td>Items</td>
<td>Approved Manufacturer List Status</td>
<td>Preferred, Alternate, Obsolete</td>
<td>User</td>
</tr>
<tr>
<td>Product Development</td>
<td>Items</td>
<td>Manufacturer Part Status</td>
<td>Active, Inactive, Pending</td>
<td>User</td>
</tr>
<tr>
<td>Product Development</td>
<td>Items</td>
<td>Manufacturer Status</td>
<td>Active, Inactive, Pending</td>
<td>User</td>
</tr>
<tr>
<td>Product Development</td>
<td>Change Order</td>
<td>Change Priorities</td>
<td>High, Low, Medium</td>
<td>User</td>
</tr>
</tbody>
</table>

Note: To add lookups in change order, run the Manage Change Priorities task in the Product Management offering (available in the Setup and Maintenance work area).

<table>
<thead>
<tr>
<th>Application</th>
<th>Module</th>
<th>Meaning (Lookup)</th>
<th>Meaning (Codes)</th>
<th>Customization Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Development</td>
<td>Change Order</td>
<td>Change Reasons</td>
<td>Quality, Safety, Cost</td>
<td>User</td>
</tr>
</tbody>
</table>

Note: To add lookups under change reasons, run the Manage Change Reasons task in the Product Management offering (available in the Setup and Maintenance work area).

| Common References | Common | Class Family Name | Design, Concept, Concept Component, | User |

---

- Revisions: Explained
Configuring Item, Document, and Change Management in Product Development: Explained

This topic explains the configuration process for items management and change management in Oracle Product Development.

To access the Product Development configuration screen, on the Tasks side tab click **Manage Configurations** (in Settings). Configuration areas include:

- **Settings Tab**
  - Default organization for items
  - Enable document management and set the root class for documents

- **Items Tab**
  - Item grading rules
  - Item lifecycle phase
  - Attributes to display in items details information tile
  - Approved manufacturer list status
  - Item attributes display settings
  - Extended attribute display

- **Change Orders Tab**
  - Cycle Time Threshold for a Change type

## Settings Tab

Use the **Settings** tab to configure the default organization, and document management.
Items Tab

Use the Grade section to configure Item Grading rules.

The Item Grading rules allow you to configure whether the BOM score in Product Development should be based on a letter or number grade. The predefined rules that you select are considered in the Item Grade score calculations.

Use the Life Cycle Phase Definitions section to configure item states that must be considered Released, Unreleased, or Obsolete, and how these item states must be displayed in the Structure information tile. When the user opens an item with a BOM, the data displayed in the information tile is based on the Released, Unreleased, and Obsolete items in the entire structure; the definition of the item lifecycle phase is calculated based on this specific setting.

In the Information Tile Details section, select three attributes you want visible in the Item Details General Information tile.

Note: No attributes are assigned to the Item Details information tile by default. To avoid configuration errors, start with defining an item template that is used to create Product Development items. This template must include the default values for attributes like Lifecycle Phase, Item Status, Primary Unit of Measure, and so on. To do this, run the Manage Item Classes task in the Product Management offering (available in the Setup and Maintenance work area). If you are using Product Development and Innovation Management together, you must also link the same default item template in the Product Development connector configuration. To do this, run the Manage Target System task in the Product Management offering (available in the Setup and Maintenance work area).

Use the Manufacturer Part Status Definitions section to classify manufacturer parts statuses and map to known system statuses such as Approved or Unapproved.

Use the Item Attributes Display Settings section to configure operational attributes that you want displayed as part of item’s general information.

Use the Extended Attributes Display section to enable or disable the display of EFF attributes.

Change Orders Tab

Use the Change Orders tab to configure the Cycle Time Threshold in Days for Change Types.

For each Change Type, define a time limit by which the change order (for that change type) must be approved. The values that you configure here are used to display unreleased changes that are within the time limit, and those past the deadline.

To create Change Types, run the Manage Change Order Types task in the Setup and Maintenance work area.

Related Topics

- Defining Product Development: Overview

Configuring Access to Supplier Portal in Product Development: Explained

The Supplier Portal is now available to Product Lifecycle Management supplier functions. Up to Cloud Release 13, it has been available to Procurement, also a Supply Chain product offering.

The supplier portal user interface is Read Only and is the same UI across product offerings. So a supplier user who works in Procurement has access to Supply Chain supplier portal with no modification.
Supplier Portal Access Role
A supplier user may have other roles that permit controlled functions other than accessing the supplier portal. For supplier portal access, the user must be assigned the Supplier Product Designer role. The supplier user does not see portal-specific tasks until the administrator has imported and activated this role for the user.

Supplier Portal Tasks
The Supplier Product Designer role controls two tasks:

- **View Items task**
- **View Change Orders task**

The View Items task permits the supplier user to gain access to an item and its details. Only items that the user is authorized to access will appear to that user.

The View Change Orders task permits the supplier user to gain access to a change order and its details. Only change orders that the user is authorized to access will appear to that user.

Oracle Social Network Objects in Product Development: Overview
This topic details the Oracle Product Development business object that you can transform to Oracle Social Network objects in Oracle SCM Cloud.

Use the **Manage Oracle Social Network Objects** task to locate the Product Development business objects and their attributes that you can enable for Oracle Social Network integration as described in the following table.

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Business Object Name</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Development</td>
<td>Change Order</td>
<td>Name, Description</td>
</tr>
<tr>
<td>Product Model</td>
<td>Item</td>
<td>Name, Description, User Item Type, Item Status, Pack Type, Primary Unit of Measure, Approval Status, Organization, Lifecycle Phase, Item, and Item Class.</td>
</tr>
</tbody>
</table>

The attributes data is sent to the Oracle Social Network at run time. If you select Manual at the time of enabling the business object, users decide whether or not to share an object instance in the social network.

Related Topics
- What are the prerequisites for Oracle Social Network integration?
- Managing Oracle Social Network Objects: Explained

Configure Integrations for Product Development
Integrating Product Development with Innovation Management

To integrate Oracle Product Development with Oracle Innovation Management, start by duplicating the ORA_PD connector template. Configure the newly created connector using the Edit option from the Actions menu. Alternatively, select the newly created connector and click the Edit icon.

Use the tasks Manage Item Organizations and Manage Item Classes in the Product Management offering, described in the following table, to create and manage item classes, item organizations, and item templates.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Default Organization</td>
<td>Select from a list of organizations in the menu.</td>
</tr>
<tr>
<td></td>
<td>Example: Vision Germany</td>
</tr>
<tr>
<td>*Default Item Class</td>
<td>Select from a list of item classes in the menu.</td>
</tr>
<tr>
<td></td>
<td>Example: Root Item Class</td>
</tr>
<tr>
<td>*Default Item Template</td>
<td>Select from the menu.</td>
</tr>
<tr>
<td></td>
<td>Example: Configured Item</td>
</tr>
</tbody>
</table>

Enabling Item Class Mapping in Product Development Connector: Worked Example

The PD connector supports flexible mapping of IM concept and component classes to PD item classes. In concept structures, you can also map specific user-defined attributes to item attributes in item structures.

The support of flexible mapping comprises these integration scenarios:

- Convert concept component to item;
- Copy item to concept component; and,
- Display item data in concept structure.

The following administrative tasks must be executed.

1. In Product Development (logged in as an administrator), open the side panel and, in Settings, click Manage Configurations. From the Settings tab, set up the Default Organization: this will be the item organization when you convert concept components to items in PD. Be sure to select the Default Organization.
2. Define an item class for each different type of component. In Innovation Management, open Setup and Maintenance. Navigate to Manage Target System. Create a new PD connector by duplicating it from the Oracle Template ORA_PD.
3. Go to Manage Mapping to External System and click the Add + button to add a subentity, that is, a subclass of the Item class. Select the new item class and add class-specific extensible flex-fields (EFFs).
Note: EFF attributes must be added on the same hierarchy level of the item class on which the EFF was defined. Also, only one connector can be active at a time.

4. In Innovation Management, go to Manage Mapping where you enable class-specific mapping for the mapping sets Create Item (CDMCreatetem), Copy Item (CDMCopyItem) and Link Item (CDMLinkitem). Select the Mapping Set to which you want to add the class mapping. Then select the entity Item and add the subclass you created.

5. Select the additional item class and add the IM attribute to Item EFF mapping (in the Attribute section). You have created a user-defined attribute for a concept structure in IM, created an EFF in an item structure in PD, and now the data can go from the concept structure to the item structure. Repeat these steps for every type/class combination to be mapped.

Note: Since a parent class must be added to the mapping before its child classes can be mapped, you may add dummy mappings for intermediate classes if they do not correspond to a concept component type. In this case, the user can type in any value for the IM Entity Name column.

Integrating Product Development with Project Management

Manage Oracle Product Development projects by associating items and change orders to a project task. Define rules to determine the completion of such tasks based on work item statuses.

Implement Oracle Product Development and Oracle Fusion Project Management for Integration

Implement the tasks listed in the following table, at minimum, in the Product Development and Product Hub functional areas in the Product Management offering.

<table>
<thead>
<tr>
<th>Functional Area</th>
<th>Tasks Enabled for Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items Organizations</td>
<td>Manage Item Organizations</td>
</tr>
<tr>
<td>Items</td>
<td>Manage Item Classes’</td>
</tr>
<tr>
<td></td>
<td>Manage Item Statuses</td>
</tr>
<tr>
<td></td>
<td>Manage Item Types</td>
</tr>
<tr>
<td>Change Orders</td>
<td>Manage Change Order Types</td>
</tr>
<tr>
<td>Product Development</td>
<td>Manage Product Development Lookups</td>
</tr>
</tbody>
</table>

After completing the integration, perform the following steps to enable the display of Work Items column in the project plan:

1. Navigate to the Setup and Maintenance work area.
2. Select the Product Management offering and click Change Feature Opt In.
3. From the View menu, select Columns > Implementation Status.
4. In the Product Development functional area, set the implementation status to Implemented.
Oracle Product Development Business Objects in Project Tasks

You can manage Product Development projects only if you are a project enterprise resource, such as project manager or team member of projects, in the project plan.

Project managers assigned the appropriate job role, such as product manager, product design manager, or product portfolio manager, can perform the following actions:

- Open and manage project work items in Product Development.
- Navigate to the Project Management work area from the Relationships tab in items and change orders, and view summary information of the related object on hover.
- Define rules to specify statuses that determine when work items can be considered complete. When a work item reaches the appropriate status, the task is updated to complete.

Related Topics

- Managing Product Development Projects: Worked Example
- Work Items: Explained
Glossary

**automatic assignment catalog**
A non-hierarchical catalog to which categories that match the catalog’s Catalog Structure value are automatically added. Add categories and share categories actions are disabled for this catalog configuration.

**catalog**
A collection of categories used to classify items which can be organized into a hierarchy that represents a taxonomy.

**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.

**descriptive flexfield**
A configurable field that captures additional information.

**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.

**GTIN**
Abbreviation for Global Trade Identification Number

**value set**
A predefined set to validate the values that a user enters in the application. The set may be hierarchical.

**work item**
An item that represents a unit of work that team members are performing on a project task. The status of the work item can determine if a task is complete.