# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>i</td>
</tr>
<tr>
<td>1 Product Management Overview</td>
<td>1</td>
</tr>
<tr>
<td>Product Management Offering: Overview</td>
<td>1</td>
</tr>
<tr>
<td>Product Lifecycle Management and Product Hub: Overview</td>
<td>1</td>
</tr>
<tr>
<td>Implementation Tasks: Overview</td>
<td>2</td>
</tr>
<tr>
<td>Setting up Innovation Management: Roadmap</td>
<td>5</td>
</tr>
<tr>
<td>Setting up Product Development: Roadmap</td>
<td>6</td>
</tr>
<tr>
<td>Setting up Product Hub: Roadmap</td>
<td>6</td>
</tr>
<tr>
<td>Setting up Product Hub Portal for Supplier Users: Roadmap</td>
<td>8</td>
</tr>
<tr>
<td>2 Defining Item Organizations</td>
<td>9</td>
</tr>
<tr>
<td>Item Organizations: Overview</td>
<td>9</td>
</tr>
<tr>
<td>Item Organizations: Explained</td>
<td>9</td>
</tr>
<tr>
<td>Organization Trees: Explained</td>
<td>10</td>
</tr>
<tr>
<td>Item Definition Organization and Reference Organization: Explained</td>
<td>10</td>
</tr>
<tr>
<td>Item Definition Organization and Reference Organization: Points to Consider</td>
<td>11</td>
</tr>
<tr>
<td>What's an item organization?</td>
<td>13</td>
</tr>
</tbody>
</table>

---

Oracle SCM Cloud
Implementing Product Management

ORACLE
# Defining Items

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define Items: Overview</td>
<td>15</td>
</tr>
<tr>
<td>Item Profile Options: Explained</td>
<td>16</td>
</tr>
<tr>
<td>Advanced Item Profile Options: Explained</td>
<td>17</td>
</tr>
<tr>
<td>Item Templates: Explained</td>
<td>18</td>
</tr>
<tr>
<td>Units of Measure, Unit of Measure Classes, and Base Units of Measure: How They Fit Together</td>
<td>19</td>
</tr>
<tr>
<td>Lifecycle Phases: Explained</td>
<td>20</td>
</tr>
<tr>
<td>Setting Up Item Lifecycle Phases: Explained</td>
<td>20</td>
</tr>
<tr>
<td>Manufacturer Parts and Manufacturers: Explained</td>
<td>21</td>
</tr>
<tr>
<td>Product Child Value Sets: Explained</td>
<td>22</td>
</tr>
<tr>
<td>Attachment Categories: Explained</td>
<td>22</td>
</tr>
<tr>
<td>Managing Attachment Security: Explained</td>
<td>22</td>
</tr>
<tr>
<td>Operational Attributes Groups: Explained</td>
<td>25</td>
</tr>
<tr>
<td>Transactional Attributes: Explained</td>
<td>26</td>
</tr>
<tr>
<td>Managing Operational Attribute Groups: Explained</td>
<td>28</td>
</tr>
<tr>
<td>Configuring Extensible Flexfields or User-Defined Attributes</td>
<td>28</td>
</tr>
<tr>
<td>Item Attribute Groups and Attributes: Explained</td>
<td>32</td>
</tr>
<tr>
<td>Attributes in Item and Change Order Context: Explained</td>
<td>32</td>
</tr>
<tr>
<td>Configure Attribute Groups for Product Development</td>
<td>33</td>
</tr>
<tr>
<td>Create Data Privileges for Attribute Groups: Procedure</td>
<td>34</td>
</tr>
<tr>
<td>Item Classes: Explained</td>
<td>34</td>
</tr>
<tr>
<td>Item Class Descriptive Flexfields: Explained</td>
<td>35</td>
</tr>
<tr>
<td>Default Item Class: Explained</td>
<td>36</td>
</tr>
<tr>
<td>Item Statuses: Explained</td>
<td>36</td>
</tr>
<tr>
<td>Setting Up Data Security for Item Classes: Explained</td>
<td>37</td>
</tr>
<tr>
<td>Managing Item Types: Explained</td>
<td>39</td>
</tr>
<tr>
<td>Cross-Reference Types: Explained</td>
<td>39</td>
</tr>
<tr>
<td>Item Descriptive Flexfields: Explained</td>
<td>40</td>
</tr>
<tr>
<td>Import Items: Explained</td>
<td>41</td>
</tr>
<tr>
<td>Related Item Subtypes: Explained</td>
<td>42</td>
</tr>
<tr>
<td>Item Revision Descriptive Flexfields: Explained</td>
<td>43</td>
</tr>
<tr>
<td>Trading Partner Item Descriptive Flexfields: Explained</td>
<td>43</td>
</tr>
<tr>
<td>Item Keyword Search Scheduled Process Actions: Explained</td>
<td>44</td>
</tr>
<tr>
<td>Managing Document Classes: Explained</td>
<td>44</td>
</tr>
<tr>
<td>FAQs for Defining Items</td>
<td>45</td>
</tr>
</tbody>
</table>
6 Defining Product Rules

Product Rules: Overview

Rules and Rule Sets: Explained

Defining Rule Sets and Item Rules: Procedures

Creating an Item Rule Set and Rules: Worked Example

Item Rule Data Types: Explained

Item Rule Syntax: Explained

Item Rule Multirow Attribute Group Functions

Item Rule Numeric Functions and Operators

Item Rule Production Value Functions

Item Rule String Functions

Item Rule Logical Functions and Operators

Item Rule Expressions with Business Entities: Explained

Item Rule Utility Functions

Attributes Supported for Item Rules

Rule Impact Analysis: Explained

Using Custom Object Data in Rules: Explained

Assignment and Validation Examples

Blending Rules

7 Defining 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

Troubleshoot Access to New Item Requests: Explained

Defining Entry and Exit Criteria for New Item Requests: Procedure

8 Defining Product Spoke Systems

Product Spoke Systems: Overview

Product Spoke Systems: Explained

9 Defining Advanced Catalogs

Advanced Catalogs: Overview

Catalog Mappings: Explained
10 **Defining Item Mass Update Configuration**

Item Mass Update Configuration: Overview

11 **Defining Imports**

- Item Import Formats Configuration: Explained
- Item Imports: Explained
- Item Data Conversion: Points to Consider
- Generate the CSV File: Explained
- Upload to the Universal Content Manager: Explained
- Import Data from the Item Management Interface Tables: Explained
- Monitor Item Imports: Explained
- Importing an Item Structure: Worked Example
- Importing Agile PLM Business Objects to Oracle Cloud: Details
- Item Import Batch Configuration: Overview
- Item Import Formats Configuration: Explained
- Importing Items with FBDI: Explained
- Import Map: Overview
- FAQs for Defining Imports

12 **Defining Audit History for Product Management**

Audit History for Product Management: Overview
Audit Trail: Explained

13 **Defining Product Lifecycle Management**

- Define Product Innovation
- Define Product Development
- Configure Integrations for Product Development
14 Defining Workflow and Approval Management

Workflow and Approval Management: Overview 169
Setting up the BPM Workflow: Explained 169
Workflow Configuration in NIRs and Change Orders: Explained 170
Configuring Tasks Using BPM Worklist: Overview 173
Configuring the Password Policy for Change Order Approval: Procedure 174
Configuration Options in the BPM Worklist 175
BPM Workflow Notifications: Overview 175
Configuring Workflow Notifications: Explained 176
Task Assignment and Routing: Explained 177
Approval Groups: Explained 178
Approval Management: Explained 179
Entry and Exit Criteria: Explained 180
Creating an Approval Rule: Worked Example 181
Routing Items to a Particular Group: Sample Rule 182
FAQs for Workflow and Approval Management 183
Preface

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

Using Oracle Applications

Using Applications Help

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

Videos included in this guide are provided as a media alternative for text-based help topics also available in this guide.
Contacting Oracle

Access to Oracle Support

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

Comments and Suggestions

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

Product Management Offering: Overview

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

- 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.

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>
</table>
| Innovation Management | • Product Requirements and Ideation Management  
|                     | • Concept Design Management  
|                     | • Product Lifecycle Portfolio Management                                                            |
| Product Development | • Product Development  
|                     | • Product Management Business Intelligence Analytics (if your users require BI Reports for Product Development) |
| Product Hub         | • Item Organizations  
|                     | • Items  
|                     | • Catalogs  
|                     | • Structures  
|                     | • Item Mass Update  
|                     | • Advanced Catalogs  
|                     | • New Item Requests  
|                     | • Change Orders  
|                     | • Product Rules  
|                     | • Audit Trail  
|                     | • Product Spoke System  
|                     | • Item Batches  
|                     | • Data Pool Integration                                                                               |

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 configuration.

<table>
<thead>
<tr>
<th>Deployment Option</th>
<th>Deployed By</th>
<th>Level of Control and Configuration</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 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. Once completed, this user can create additional users for the rest of the implementation team.</td>
</tr>
</tbody>
</table>

**Tip:** 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.

To create the primary user, follow these instructions:

1. Sign in with your 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.
6. To provision the new user with roles, search for the Application Implementation Consultant role. Select it and click Add Role Membership.
7. Add the following Roles, at minimum:
   - IT Security Manager
   - Employee

Close the window.

Notify your primary implementation team member that their user ID has been created. Give them their initial password.

| **Step 2- Create Implementation Projects** | Optionally, create Implementation Projects in the Setup and Maintenance work area. |

| **Step 3- Set up Key Implementation Users and Security Profiles** | After your environments are provisioned, you as the Service Administrator have sufficient security abilities to create three implementation users with the necessary roles. Examples: |

- TechAdmin: Can perform key technical duties, including functional setup and assigning security roles to users.
- APPL_IMPL_CONSULTANT and SCM_IMPL_CONSULTANT: Can perform key functional duties, including functional setup. |
Tasks for the Service Administrator | Description
--- | ---
You may decide to replace or refine these initial users, but these users have all the access required to get you started.

Tasks for Application Implementation Consultant | Description
--- | ---
**Step 4- Create Data Roles and Assign Security Profiles**
You can secure data by provisioning roles that provide the necessary access rights.

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 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.
Implementing Product Management

Chapter 1

Product Management Overview

Tasks for Application Implementation Consultant

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 6- Perform Common Application Configuration</td>
</tr>
</tbody>
</table>

For more detailed information on common implementation tasks for all SCM products, see Oracle Applications Cloud Implementing Common Features for SCM guide.

Related Topics

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

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>

Define Concept Design Management

Use this task list to configure concepts and concept components.

- Manage Product Concept Classes
- Manage Product Concept Component Classes
- Manage Product Concept Statuses
- Manage Product Concept Lookups

Define Product Lifecycle Portfolio Management

Use this task list to configure proposals and portfolios.

- Manage Proposal Statuses
- Manage Product Portfolio Classes
### 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:

<table>
<thead>
<tr>
<th>Task List</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Manage Item Organizations</td>
<td>Use these tasks to define item organizations for Product Management.</td>
</tr>
<tr>
<td>• Manage Organization Trees</td>
<td></td>
</tr>
<tr>
<td>• Manage Item Classes</td>
<td>Use these tasks to define items.</td>
</tr>
<tr>
<td>• Manage Item Attribute Groups and Attributes</td>
<td></td>
</tr>
<tr>
<td>• Manage Lifecycle Phases</td>
<td></td>
</tr>
<tr>
<td>• Manage Change Order Types</td>
<td>Use these tasks to define change orders.</td>
</tr>
<tr>
<td>• Manage Change Order and New Item Request Header Descriptive Flexfields</td>
<td></td>
</tr>
<tr>
<td>• Manage Change Order Entry and Exit Rule Sets</td>
<td></td>
</tr>
<tr>
<td>• Manage Structure Component Descriptive Flexfields</td>
<td>Use these tasks to define structure component and lookups.</td>
</tr>
<tr>
<td>• Manage Product Development Lookups</td>
<td></td>
</tr>
</tbody>
</table>

### 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>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>
</tbody>
</table>
### Task List

<table>
<thead>
<tr>
<th>Task List</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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. For more information, see the Oracle Procurement Cloud Implementing Procurement guide.
2 Defining 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 supply chain management 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 Applications Cloud. 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 the logistics applications) 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.

Item Definition Organization and Reference Organization: Explained

An item definition organization is an item organization that stores the item attribute values. A reference organization is an item organization that collects the item attribute values from the item definition organization.

Currently, you have to model every location as an organization. Most of the item attributes are similar within a group of organizations. Typically, all locations within a group have the exact values for the item attributes. For example, if one million items are present in the Item Master, several million records are stored in the database. You create multiple records in the database and most of the data have duplicate records. It increases the time to query a record and it becomes difficult to maintain the database. To avoid this situation, you can model your organizations into two groups, namely item definition organization and reference organization. By doing so, the attribute values of items in the reference organization are derived from the item definition organization. This method reduces the total number of records in the database.

You can define the item definition organization on the Manage Inventory Organization Parameters setup page. You can define the reference organization using the Item Grouping Behavior attribute on the Manage Inventory Organization Parameters setup page.

Benefits of Modeling Definition Organizations

When you model your organizations into definition organizations and reference organizations, you have the following benefits:

- Reduction in the data migration time
• Reduction in item data volume for large, complex organizations
• Reduction in data maintenance effort. Any attribute change to a definition organization is available to all the reference organizations automatically.
• Reduction in the number of item rules
• Reduction in the data audit effort. You need to audit only the definition organizations.
• Improvement in the performance of item query, keyword searches, and transactions for large, complex organizations

Item Definition Organization and Reference Organization: Points to Consider

When you create an item definition organization, you must consider how you want to manage the organization in your item-organization setup. In a reference organization, all item data are read-only, except packs. All item data comes from the associated definition organization. The following list includes some of the item data that are read-only:

• Attachments
• Attributes
• Category and Category Assignments
• GTIN
• Item Structures
• Source System Cross References
• Spoke Systems
• Trading Partner Items

Consider the following points related to item definition organizations and reference organizations when you:

• Create an organization
• Update an existing organization
• Convert an organization to a definition or a reference organization
• Set up a manufacturing plant

Creating an Organization

Consider the following points when you create an organization:

• The default value for the Item Definition Organization field is the value that you select in the Master Organization field.
• If the item attributes do not vary for the newly created organization, then select the organization from where you would reference the item records as the value for the item definition organization.
• If your business requires item attributes to vary in the newly created organization, then use the same organization as the item definition organization. For example, if you create Vision Germany as the new organization for your business, then set Vision Germany as the item definition organization.
• When you create a new reference organization, associate an existing item definition organization with the reference organization. After the association, a new row is created in the EGP_ITEM_ORG_ASSOCIATIONS table for every item that is assigned to its item definition organization.
Upgrading Existing Organizations
When you upgrade from a previous release to a new release, consider the following points related to organizations:

- Update all master and child organizations of item organizations, and materials management organization in SCM, and define the item definition organizations to itself.
- Update existing item definition organization to a reference organization. This will reduce the records in the EGP_SYSTEM_ITEMS_B table.

Converting Definition Organization to Reference Organization
To convert a definition organization to a reference organization, the organizations and items must meet certain conditions. If the conditions are not met, then the validation checks result in an error and the conversion fails.

You cannot convert a definition organization to a reference organization if any of the following conditions are met:

- The revision code or effective date for an item are different in the reference and definition organizations
- The supplier-organization association for an item is different in the reference and definition organizations
- The item-category assignments for organization-controlled catalogs are different in the reference and definition organizations
- The status of change order for an item is anything other than Completed or Failed
- The structures are common from an organization other than definition organization.
- The definition organization is a manufacturing plant.

Converting Reference Organization to Definition Organization
You can convert a reference organization to a definition organization and maintain a copy of the item record in the new definition organization. No validations are run for this conversion.

Changing Definition Organization of Associated Reference Organizations
You can convert a definition organization to a reference organization only after you have realigned all its associated reference organizations to another definition organization. You can also convert all the associated reference organizations to a definition organization. If you do not reassign the associated reference organization before the conversion, you will receive error messages depending on the validation checks, and the conversion will fail.

Setting Up a Manufacturing Plant
Consider the following points when you set up a manufacturing plant:

- You can define a new manufacturing plant either as a definition organization or a reference organization. However, if you are looking for the following capabilities in your organization, then do not set up a manufacturing plant as a reference organization:
  - You want to calculate manufacturing lead times and update the lead-time values for items.
  - You want organization-specific supply subinventory for an item.
  - You want organization-specific serial-control attributes and lot-control attributes for an item.
You can convert a manufacturing plant that is a reference organization to a definition organization. However, you cannot convert a manufacturing plant that is a definition organization to a reference organization.

What's an item organization?

Item organizations contain only definitions of items. Use item organizations in implementations when the storage or movement of inventory does not need to be physically or financially tracked. For example, in a retail implementation you can create an item organization to contain only the definitions of items that are listed by and sold through each retail outlet, while a different system tracks the physical inventory and transactions of those items. If Oracle Fusion Inventory Management is installed, you can change an item organization to an inventory organization.
3 Defining 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 Management.</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 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.
**Item Templates: Explained**

Use item templates to assign attribute values during item creation in Oracle Fusion Product Hub.

An item template is a defined set of attribute values. When you apply an item template to an item in an organization, you insert the attribute values into the item definition. The defined attribute values from the item template automatically populate the appropriate item attribute fields.

Item templates are specific to an item class and an organization. The templates are inherited in a hierarchical item class and apply to all child item classes. Item templates can contain values for predefined attributes and user-defined attributes.

In addition to predefined item templates, you can create item templates, if required. You can also copy existing item templates to multiple organizations.

**Create Item Templates**

You can select a predefined item template to apply to an item. You can also create item templates in an organization. To create item templates, use the Manage Item Classes task available for the Product Management functional area in the Setup and Maintenance work area.

You can also edit or delete item templates as required.

**Copy Item Templates**

You can copy multiple existing item templates to multiple organizations simultaneously.

Copy the item templates as follows:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup and Maintenance page, click the **Product Management** offering.
3. On the Setup: Product Management page, click the **Items** functional area, and then click the **Manage Item Classes** task.
4. Select the item class with the item template to copy, and click **Edit**.
5. In the Templates and Formats tab of the Edit Item Class page, select the item templates to copy, and click **Copy**.
6. In the Copy Templates dialog box, search and select the organizations where the templates are to be copied, and click **OK**. You can also search and select an organization hierarchy, in which case the template will be copied to all organizations in the hierarchy.

You can specify a default item template in an organization. To do so, select the **Default** check box for an item template in the Templates and Formats tab. The item template is used by default when you create an item.

**Add Predefined Item Templates**

If the predefined item templates do not automatically appear in an organization, you can add them manually.

Add the item templates as follows:

1. In the Edit Item Class page, click **Actions > Add Predefined Templates**.
2. In the Add Predefined Templates dialog box, search and select the organization in which to add the item templates, and click **OK**.

**Related Topics**
- **Items: Explained**

## 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 following figure 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.
Setting Up Item Lifecycle Phases: Explained

Item lifecycle phases represent the stages that an item or product goes through in its lifecycle. For example, for an item that can be used in production, the representative lifecycle phase is Production. 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.

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

Lifecycle phases are associated with item classes, and the items in an item class can be assigned to any of the lifecycle phases associated with that item class. Before you create or import items, you must create lifecycle phases and assign those phases to item classes. 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.

To set up lifecycle phases:

1. In the Setup and Maintenance work area, select the Product Management offering.
2. On the Setup: Product Management page, search for and click the Manage Lifecycle Phases task.
3. Click the Create Lifecycle Phase button.
4. Enter values in the following fields:
   a. Sequence
   b. Phase (name of lifecycle phase)
   c. Phase Code
   d. Select Phase Type
5. Save and close the Manage Lifecycle Phases page.
6. Associate the lifecycle phase with one or more item classes.
   a. In the Setup and Maintenance work area, select the Product Management offering.
   b. On the Setup: Product Management page, search for and click the Manage Default Item Class task.
   c. Click on the Lifecycle Phases Tab. This tab contains all lifecycle phases that are associated with the root item class.
   d. Associate the lifecycle phase to the root item class.
      i. Click the Lifecycle Phase tab.
      ii. Click on Add Icon. The Search and Add: Lifecycle Phases dialog is displayed.
      iii. Search for the Lifecycle Phase that you created.
      iv. Click the OK button to associate the lifecycle phase with the item class.
7. Click the Save and Close button to save the changes.

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.

Related Topics
- Attachments: Explained
- Attachment Entities: Explained
- Attachment Entities and Attachment Categories: How They Work Together
- What’s an attachment category
Managing Attachment Security: Explained

The privileges for accessing the attachments of an item are by default inherited from business objects. You can further define the security of item attachments so that different users can have access to the same item but only have access to certain categories of attachments to that item. For example, there may be multiple categories of objects (such as specifications, drawings, or financial documents) attached to an item. There may be multiple job roles (such as buyer, design engineer, or accountant) who have access privileges to the item. Though they all may have the same access to the item itself, their access to the attached objects may need to be restricted by attachment category. By default, all job roles are granted access to the predefined attachment category Miscellaneous.

Any security policy defined for an attachment category is enforced on all business objects to which the attachment category is associated only if those business objects are enabled for attachment security.

To provide attachment security, perform the following tasks:

- These tasks are not specific to attachment security, but are required prerequisites, to be performed once for each attachment category:
  - Create attachment categories.
  - Assign the attachment categories to item classes.

- These tasks are specific to attachment security:
  - Define data security policies, which apply to attachment categories.
  - Enable data security policies for selected business objects.

Creating Attachment Categories

Attachment data security is implemented by using attachment categories. Attachment categories affect access to attachments through the item classes for the items being edited. Attachment data security can be assigned individually at the user level. It can also be assigned at the group level through job roles. You create attachment categories with the Manage Attachment Categories task in the Setup and Maintenance work area. You associate each attachment category with attachment entities that represent business objects: items, item revisions, catalogs, categories, and trading partner items.

Assigning Attachment Categories to Item Classes

You assign attachment categories to item classes with the Manage Item Classes task in the Setup and Maintenance work area. While editing an item class, you associate it with one or more attachment categories for which you want to provide security. This association is required only for attachment categories that are associated with attachment entities at the item level and item revision level. Since attachment categories are inherited down through the item class hierarchy, you can associate an attachment category with all item classes by assigning it to the Root Item Class.

Defining Data Security Policies

A data security policy is defined by a set of allowable actions on a database resource (such as an attachment category) for a job role. When that role is provisioned to a user, the user has access to the data defined by the policy. That is, an attachment data security policy defines who (defined as a job role) can perform what operations (such as read, update, or delete) on which set of attachment categories, according to a defined condition.
To define a data security policy for an attachment category:

1. Sign in with the predefined IT Security Manager role.
2. In the Setup and Maintenance work area, select the Product Management offering.
3. Search for the Manage Data Security Policies task, then select and open the task. This task may be available to you under more than one functional area. The Security Console is automatically launched by the Manage Data Security Policies task.

You can also open the Security Console directly from the Navigator.

4. On the General subtab of the Administration tab of the Security Console, click Manage Database Resources.
A database resource defines an instance of a data object. A data object is a table, view, or flexfield.

5. On the Manage Database Resources and Policies page, search for the Display Name equal to Application Attachment Category. The category appears in the search results, with an Object Name of FND_DOCUMENT_CATEGORIES

The data security policies defined for the selected database resource appear in the Policies Details region.

6. In the Search Results region for the selected database resource, select Edit from the Actions menu.

7. On the Condition tab of the Edit Data Security page, select Create from the Actions menu.

8. In the Create Database Resource Condition dialog box, name the condition and specify the attachment categories in scope for the data security policy.

The following table suggests values for an example condition:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>IPDrawings</td>
</tr>
<tr>
<td>Display Name</td>
<td>IPDrawings</td>
</tr>
<tr>
<td>Description</td>
<td>IP Drawings-Restricted to R&amp;D</td>
</tr>
<tr>
<td>Condition Type</td>
<td>SQL predicate</td>
</tr>
<tr>
<td>SQL Predicate</td>
<td>The SQL predicate consists of a query on the table or view named by the database resource (in this example, FND DOCUMENT CATEGORIES). The category name specified in the predicate must exactly match the name that you specified when you created the attachment category.</td>
</tr>
</tbody>
</table>


10. On the General Information tab of the Create Policy dialog box, specify the module. By default, the Module field is the module associated with the database resource for which you’re creating the policy.

11. On the Role tab of the Create Policy dialog box, select fscm in the Application list, then search for and select the role names to be assigned the new policy, such as Product Data Steward.

12. On the Rule tab of the Create Policy dialog box, select Multiple Values in the Row Set field, then search for and select in the Condition field for the name of the condition that you created, such as the example here, Secured Attachments for Product Hub.

13. On the Action tab of the Create Policy dialog box, move actions from the Available Actions list to the Selected Actions list to specify the actions that are applicable to the data secured on the database resource, which you want to grant to the roles you selected.
14. On the Edit Data Security page, click **Submit** to update the database resource **FND_DOCUMENT_CATEGORIES**.

15. On the Manage Database Resources and Policies page, click **Done**.

### Enabling Attachment Data Security for Business Objects

You can enable and disable attachment security at the level of business objects. When you enable attachment security for a specific business object, then attachment security is enforced for every attachment category assigned to the business object. Note that, by default, all job roles are granted access to the predefined attachment category **Miscellaneous**.

To enable your data security policies on attachment categories:

1. Sign in with the SCM implementation consultant role.
2. In the Setup and Maintenance work area, select the Product Management offering.
3. Search for and open the Manage Applications Core Attachment Entities task.
4. On the Manage Attachment Entities page, you will search for and select each of the attachment entities that you previously assigned to the attachment categories that you created. Attachment entities represent business objects: items, item revisions, catalogs, categories, and trading partner items.

Enter one of the following attachment entity names in the **Entity Name** field and click **Search**. The attachment category that you created should appear in the Attachment Categories region for the selected attachment entity.

<table>
<thead>
<tr>
<th>Business Object Attachment Association Level</th>
<th>Attachment Entity Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item Level</td>
<td>ITEM_ENTITY</td>
</tr>
<tr>
<td>Item Revision Level</td>
<td>ITEM_REVISION_ENTITY</td>
</tr>
<tr>
<td>Trading Partner Level</td>
<td>EGP_TRADING_PARTNER_ITEMS</td>
</tr>
<tr>
<td>Catalog Level</td>
<td>CATALOG_ENTITY</td>
</tr>
<tr>
<td>Category Level</td>
<td>CATEGORY_ENTITY</td>
</tr>
</tbody>
</table>

5. For each selected attachment entity in the search results, click **Enable Security**.
6. When you have enabled security on all the desired attachment entities, click **Save and Close**.
7. On the Setup page, search for and open the Run User and Roles Synchronization Process task from the Initial Users functional area.
8. Submit the scheduled process to complete enabling security on attachments.

**Related Topics**

- Attachment Entities: Explained
- Attachment Entities and Attachment Categories: How They Work Together
- What's an attachment category
- Database Resources and Data Security Policies: How They Work Together
- Securing Attachments
Operational Attributes Groups: Explained

Operational attributes determine the behavior of the item with respect to various applications outside of Oracle Fusion Product Hub, such as Oracle Fusion Purchasing or Oracle Fusion Inventory Management.

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.

Transactional Attributes: Explained

Attributes that exist for each instance of an item and the values for the attributes can be different.

For example:

- The number of megabytes (MB) or gigabytes (GB) of e-mail storage on a digital subscriber line account.
- The monogram text on a shirt pocket.
- The color of a music player.

These attributes are defined at the item class and their attribute value is captured at the time of a transaction by downstream applications. The metadata values of these attributes are maintained at the item class. Order orchestration and order capture systems are two examples of downstream use. All transactional attributes must be associated with a value set.

The following metadata values can be defined for an attribute.

- Required: Indicates whether the attribute value is required at the transaction.
- Default Value: Indicates the default value of the attribute.
- Value Set: Indicates the value set associated with the attribute.
- Read Only: Indicates whether the attribute value is read only.
- Hidden: Indicates whether the attribute is not shown.
- Active: Indicates whether the attribute is active or inactive.
Transactional attributes are inherited across the item class hierarchy. The metadata is data-effective. Changes in the metadata will be reflected immediately at the item level. For example:

- Any of the metadata of a transactional item attribute belonging to a specific domain, if modified in the child item class would break the inheritance. Any changes done at the parent item class for this transactional item attribute would not get inherited. Multiple records with same date range can exist if they belong to different domains. For example, the transactional item attribute *Memory* is associated with a *Domain* and order capture. Each of the domains may use a different set of metadata for its own purpose. Hence, for the same date range, two different records can exist. Only Start Dates for a transactional item attribute would be entered by a user. End date would be calculated automatically based on the next Date Effective record.
- Users can modify (either Start Date and metadata) of a future effective record. Records with Starting date as Past cannot be modify or edited.
- Only start dates can be set to permit updating by a user, and the end date of a record will automatically be pulled from the next record.
- Any changes performed in the parent item class would be inherited by the child item class. If the corresponding record is modified in the child, then these changes will not be inherited.

Item pages provide a mechanism with which to configure the user interface.

### Pages and Attribute Groups

Pages and attribute groups enable you to structure your data.

Benefits include:

- You can combine and sequence attribute groups into pages.
- There is no limit on the number of attribute groups associated with a page.
- Pages can be created at item class and are inherited down the item class hierarchy.
- Attribute groups can be added to pages sequentially and based on this sequence, these attribute groups are shown in items.
- Attributes groups can be added for an inherited page at the child item class.

Functional Item pages are another type of special pages which are used to associate pages already created for use in the application. Application scope indicates the application which uses these pages and the usage indicates the specific use of the configured pages.

### Data Quality

You can associate attributes for the purpose of standardization and matching, to be performed when items are created. You restrict the attributes to be processed for standardization or matching or both. Selecting Standardization allows the data quality engine to return the standardized values for these attributes. Matching allows the data quality engine to return any existing items which matches the value of these attributes and are potential duplicates.

### Lifecycle Phases

Sequential lifecycle phases enable you to track and control the lifecycle phases of items. Each phase represents a set of tasks and deliverables that are required before promoting the item to the next phase. You can associate lifecycle phases to an item class which are created elsewhere. Lifecycle phases are inherited down the item class hierarchy and new lifecycle phases can be added to child item classes. For example, the lifecycle phases for a computer component item class might be: Concept, Prototype, Production, and Retirement.
Templates

Template is a defined set of attribute values used during item creation. When you apply a template to an item, you overlay or default-in the set of attribute values to the item definition. For example, every time users in a particular organization create new items, the attributes, as defined and approved by the organization appear in the appropriate fields. No user guesswork is required, and time is saved during the creation of items with a similar form, fit and function. Templates are created for each item class. Templates are specific to organization. Templates are inherited down the item class hierarchy. You can define both operational attributes and user defined attributes for each template.

Search and Display Format

Search formats provide a convenient way to save frequently used search criteria. Search formats created at item class will be available to all users. Search formats are always created in the context of item class. Display formats enable you to predefine search display views. You can use these views to look at different sets of item attributes that are returned by the search. Display formats created at item class will be available to all users. Display formats are always created in the context of item class.

Import Format

An import format identifies the base and user-defined attributes in an item class that are imported into the application using a spreadsheet. Consequently, when you import item business entities from a spreadsheet, the items are all imported into the particular item class defined in the import format. These imported item business entities inherit all the attribute groups defined for the specific item class. You cannot edit the layout of an import format once it is created.

Related Topics

- Item Attributes: Explained

Managing Operational Attribute Groups: Explained

You view operational attribute groups and define their control level for the attributes associated with them on the Manage Operational Attribute Groups page.

Attribute groups are listed in the Operational Attribute Groups table. Select an attribute group, and attributes associated with it are listed in a lower table.

In the Controlled At column, you define the organization level (master and child) at which selected attribute will be controlled.

Configuring Extensible Flexfields or User-Defined Attributes

User-Defined Item Attributes and Attribute Groups: Explained

User-defined attributes are based on extensible flexfields. You create an attribute group, which determines which attributes are used at runtime. Extensible flexfields or user-defined attributes are not available to customers who only license Product Model.
User defined attributes can have a static or dynamic list of valid values, or a range of values. Values for user-defined attributes are defined when you create the item and remain the same for the life cycle of the item.

User-defined attributes can be saved within attribute groups. You associate user-defined attributes with items by adding attribute groups to item classes.

For example, items that are part of the Small Gasoline Engines item class include the following specifications:

- RPM
- Power
- Oil and fuel mixture
- Weight

An attribute is defined for each of these specifications and these attributes are grouped together as the Engine Performance attribute group.

You create attribute groups on the Manage Attribute Groups page.

**Note:** For each user-defined attribute, you can optionally define validation rules to be applied when the user inputs data.

An attribute group can be single-row, multi-row, or variant.

Multi-row attribute groups enable you to associate multiple sets of attribute values with the same object instance. It gives you the ability to store multiple additional attributes that are dependent on the item, organization, attribute group, and a unique value within the attribute group.

For example, if your item is a book, you can create an attribute group named Chapters containing the following attributes:

- Chapter number
- Name
- Number of pages

Multiple rows of Chapters can be associated with a book, while Name and Number of pages each require a single row. The attribute Chapter number is identified as Part of Unique Key.

After you create the attribute group and attributes, perform the following tasks to complete the extensible flexfield setup:

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

**Tip:** Sets of user defined attribute groups can be organized on a single page that can then be linked from the Specifications tab.

**Related Topics**

- Extensible Flexfields: Explained
- Item Specifications and Attributes: Explained
- Additional Item Attributes: Explained
Create Extensible Flexfields or User-Defined Attributes: Procedure

You can create user-defined attributes or extensible flexfields from the Product Management offering in the Setup and Maintenance work area.

To create extensible flexfields, follow these steps:

1. Create an attribute group.
2. Create attributes within the group and a corresponding value set for each attribute.
3. Deploy the item attributes.
4. Associate the attribute group to the item class.
5. Create pages and link attribute groups to pages.
6. Deploy the item attributes.

You can now view the attribute groups in the item master record. Each of these steps are described in detail in separate topics.

For more information on creating extensible flexfield, see User Defined Attribute Groups and Attributes (Extensible Flexfields-EFFs) Setup White Paper (Doc ID 1992317.1) on My Oracle Support at https://support.oracle.com.

Create Item Attribute Groups and Attributes: Procedure

Create item attribute groups and attributes using the Manage Item Attribute Groups and Attributes task from the Setup and Maintenance work area.

Follow these steps to create item attribute groups:

1. From the Setup and Maintenance work area, click the Product Management offering.
2. Search and open the Manage Item Attribute Groups and Attributes task.
3. On the Manage Attribute Groups page, create an attribute group and specify its context usage as item.

You can provide the attribute group with view or edit privileges, or both. For each user-defined attribute, you can optionally define validation rules to be applied when the user enters any data.

> **Note:** When creating or editing attribute groups, or when creating new item classes associated with attribute groups, you must redeploy the extensible flexfield so that the attribute group is available in the application. To redeploy the flexfield, search for the flexfield code named EGO_ITEM_EFF in the Manage Attribute Groups page. In the search results, select the row that contains the flexfield code EGO_ITEM_EFF and click Deploy Flexfield. After the flexfield deployment is complete, click OK. Check the Deployment Status and Deployment Date columns to verify that the extensible flexfield deployed successfully on the current date.

Associate Attribute Groups and Pages to Item Classes: Procedure

Pages include a collection of attributes groups and attributes. You can add multiple attribute groups to a page and specify the display sequence of attribute groups. The page is then associated with an item class so that the attributes are visible on the item page. You can add multiple pages to an item class. Child classes inherit the attribute groups that are created for a parent item class. Additionally, you can create attribute groups specific to a child class.
In order to add an attribute group to a page, you must add the attribute group to the attribute groups tab in the item class. After you add the attribute group to the attribute groups tab, you must save the item class page.

Follow these steps to associate attribute groups and pages to an item class:

1. Open the **Manage Item Classes** task from the Product Management offering in the Setup and Maintenance work area.
2. Search for an item class in which you want to configure attribute groups.
   - In the Pages and Attributes tab, select **Attribute Groups**.
   - Create new attribute groups or modify the existing attribute groups, as required.
   - Save the item class page.
3. In the Pages and Attributes tab, select **Pages**.
   - Select the page for which you want to configure attribute groups or create a new page.
   - Assign an attribute group to the page.
   - Specify the sequence of attribute groups.
4. In the Pages and Attributes tab, select **Functional Item Pages**.
   - Select the functional area and save the item class page.

**Note:** Functional item pages are mostly used by other products that consume extensible flexfields. For example, you can use the functional item pages to define extensible flexfields for Product Development.

You have associated the user-defined attribute groups (extensible flexfields) and pages to the item class. To view the user-defined attributes in the item page, you must deploy the extensible flexfields.

### Deploy Item 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 Flexfields** task in Setup and Maintenance work area. All 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 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 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 flexfield configuration on the item.
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 attribute, 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 and specify its context usage as item. You can provide the attribute group with view or edit privileges, or both. 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.

When you configure an attribute group, you can assign privileges. Using these privileges you control the user’s ability to view or edit attributes in the following: view or edit Item tasks, change order impact analysis, and item structure report. To do this, under Context Usage select the view and edit privileges for the 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 Flexfields task

**Related Topics**

- Descriptive Flexfields: Explained
- Product Value Sets: Explained

Attributes in Item and Change Order Context: Explained

Generally, an item attribute can be edited through the item page or change order page. However you can restrict the attribute to be edited only through change order page, by using the Manage Item Rule Set task in the Setup and Maintenance work area. You can access this task from the Product Management offering and the Product Rules functional area.

This table shows the properties of attribute types in item page and change order context.
Configure Attribute Groups for Product Development

For Product Development you can configure the attributes groups displayed as part of item, document, and change order. You specify the attribute groups for an item class and set the display sequence of attribute groups. This configuration affects attribute groups appearing on: item details page, create item dialog, change order affected object, change order impact analysis, and item structure report. Attribute groups created for an item class at a higher level are inherited to child classes. Additionally, you can create attribute groups specific to a child class.

Configure attribute groups as follows:

1. In the Setup and Maintenance work area, go to the following:
   - Offering: Product Management
   - Functional Area: Items
   - Task: Manage Item Classes

2. Search for an item class in which you want to configure attribute groups.
   - In the Pages and Attributes tab, select Attribute Groups.
   - Create new attribute groups or modify the existing attribute groups, as required.

3. In the Pages and Attributes tab, select Pages.
   - Select the page for which you want to configure attribute groups or create a new page.
   - Assign an attribute group to the page. Also specify the sequence of attribute groups.

4. In the Pages and Attributes tab, select Functional Item Pages.
   - Select the page you created or modified in the previous step and assign it to Product Development.

### Attribute Type | In Item Page Context | In Change Order Context
--- | --- | ---
Item attributes controlled by change order | Editable only in the initial item revision and if the item is not yet assigned to change order. | Editable.

| Attribute Type | In Item Page Context | In Change Order Context |
--- | --- | ---
Item attributes that are not controlled by change order | The following revisions are editable:  
- Revision that is already effective.  
- Revision that is scheduled to be effective in future.  
All revisions display the value which is already effective; even previous revisions do not display revision specific value. | Editable.  
All revisions display change order specific values.  

**Note:** The previously effective revisions are not editable even during execution of ESS job.
You must deploy item flexfields after you create or modify attributes.

Create Data Privileges for Attribute Groups: Procedure

User defined attribute groups are supported by following tables at the following data levels:

Non-translatable tables:
- Item Data Level - EGO_ITEM_EFF_B
- Item Revision Data Level - EGO_ITEM_REVISION_EFF_B
- Item Supplier Data Level - EGO_ITEM_SUPPLIER_EFF_B

Translatable tables:
- Item Data Level Translatable - EGO_ITEM_EFF_VL
- Item Revision Data Level Translatable - EGO_ITEM_REVISION_EFF_VL
- Item Supplier Data Level Translatable - EGO_ITEM_SUPPLIER_EFF_VL

To create data privileges for attribute groups:

2. Click Manage Database Resources.
3. Enter the criteria to search for an object (or database resource) and click Search.
4. Select object for which you want to create data privileges.
5. From the Actions menu, select Edit.
   The Edit Data Security page appears.
6. Click the Actions tab.
7. Click Add to create a new data privilege.
8. Enter the name of the data privilege, display name and description.
9. Click Submit.

If the Manage Database Resources button is disabled, follow these instructions to enable the button:

2. Search for the Security Management role (ORA_ASE_SECURITY_MANAGEMENT_DUTY) and edit that role.
5. Click Add Privilege to Role and click Next.
6. In Summary and Impact Report, click Save.

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: 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 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.

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.

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

## Setting Up Data Security for Item Classes: Explained

Before you can create or view items in the Product Information Management work area, you must define data security for items. Define data security in the item class for each role/organization pairing and user/organization pairing.

Initially, you must define data security in the root item class and the master organization for both the Product Manager and the Product Data Steward roles. If you created an implementation user to create items in the Product Information work area, then you must assign the Product Manager and Product Data Steward roles to that user, and you must assign that user to the master organization. (Assign the appropriate role or roles and organizations to any additional users you create to control what each user is allowed to do in the application.) You can assign all of the actions to the Product Manager and Product Data Steward role for the master organization to allow all users with these roles to have complete access to item data.

The following table describes the specific actions that you must assign to both the Product Manager and the Product Data Steward roles for the root item class and the master organization you created. (When you create additional organizations, you must define data security for each organization that these roles are assigned to.)

<table>
<thead>
<tr>
<th>Actions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintain Item Asset Maintenance Group</td>
<td>Allows access to edit item asset management specifications. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item Attribute</td>
<td>Allows access to edit item user defined attribute specifications. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item Basic</td>
<td>Allows access to edit item basic information including attachments, organizations, suppliers, relationships, and other related information.</td>
</tr>
<tr>
<td>Maintain Item Costing Group</td>
<td>Allows access to edit item costing specifications. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item General Planning Group</td>
<td>Allows access to edit item general planning specifications. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item Inventory Group</td>
<td>Allows access to edit item inventory specifications. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item Invoicing Group</td>
<td>Allows access to edit item invoicing specifications. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item Lead Times Group</td>
<td>Allows access to edit item lead times specifications. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item MRP And MPS Group</td>
<td>Allows access to edit item MRP and MPS specifications. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item Order Management Group</td>
<td>Allows access to edit item order management specifications. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item Pack</td>
<td>Allows access to edit item packs. Does not encompass view privilege.</td>
</tr>
</tbody>
</table>
## Defining Items

<table>
<thead>
<tr>
<th>Actions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintain Item People</td>
<td>Allows management of user access to items in the enterprise. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item Physical Group</td>
<td>Allows access to edit item physical specifications. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item Primary Group</td>
<td>Allows access to edit item primary specifications. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item Process Manufacturing Group</td>
<td>Allows access to edit item process manufacturing specifications. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item Purchasing Group</td>
<td>Allows access to edit item purchasing specifications. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item Receiving Group</td>
<td>Allows access to edit item receiving specifications. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item Revision</td>
<td>Allows access to create and manage item revisions. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item Service Group</td>
<td>Allows access to edit item service specifications. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item Structure</td>
<td>Allows access to create and manage item structures. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item Structure Group</td>
<td>Allows access to edit item structure specifications. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item Web Option Group</td>
<td>Allows access to edit item web option specifications. Does not encompass view privilege.</td>
</tr>
<tr>
<td>Maintain Item Work In Process Group</td>
<td>Allows access to edit item work in process specifications. Does not encompass view privilege.</td>
</tr>
<tr>
<td>View Item Attribute</td>
<td>Allows access to view item user-defined attribute specifications.</td>
</tr>
<tr>
<td>View Item Basic</td>
<td>Allows access to query and view item basic information including attributes, attachments, organizations, suppliers, and relationships.</td>
</tr>
<tr>
<td>View Item Pack</td>
<td>Allows access to view item packs.</td>
</tr>
<tr>
<td>View Item Structure</td>
<td>Allows access to view item structures.</td>
</tr>
<tr>
<td>Create Item Class Item</td>
<td>Allows access to create items within an item class.</td>
</tr>
<tr>
<td>Read</td>
<td>Read</td>
</tr>
<tr>
<td>Update</td>
<td>Update</td>
</tr>
</tbody>
</table>
For initial start up, define the data security at the root item class level and define the complete set of actions for the person or groups. Data security defined at the root item class level is inherited by all new item classes created. To define data security for an item class and organization:

1. In your implementation project, search for the Define Advanced Items task list.
2. Click the Go to Task icon for the Manage Item Classes task.
3. Select the Root Item Class row and click on the Edit icon.
4. Click on the Security tab. The Item Class People and Actions table are initially empty.
5. To add a new row, click the Add icon in the Item Class People table.
6. In the Group field, choose Principal.
7. In the Name column, click on the Search link.
8. Enter Product for the role name and click on the Search button. The results show all combinations of the roles Product Manager or Product Data Steward and the organizations to which they were assigned. In addition, two of the rows have no organization assignments.
9. Select the Product Data Steward row without an organization assignment and click the OK button.
10. Select the organization that you created in the Creating Item Organizations: Explained section of this document.
11. Define the actions that the Product Data Steward and Product Manager Roles can perform by adding actions to the Actions table.
   a. Click on the Add icon to launch the Select and Add Action dialog.
   b. Perform a search for each of the following terms, select all of the returned actions, then click apply:
      - Maintain
      - View
      - Create
   c. After selecting all of the appropriate actions, click OK to close the dialog.
12. Repeat this process to define item class security for the Product Manager.
13. Save your changes.

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 Hub, 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>
</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

Import Items: Explained
• Item revisions
• Item category assignments
• Item associations*
• Item relationships
• Item flexfields *
• Item translatable flexfields*
• Item revision flexfields*
• Item revision translatable flexfields*
• Item supplier flexfields*
• Item translatable supplier flexfields*
• Item style variant attribute value Sets
• Trading partner items

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

The following is an overview of the item import process:

1. Download the item import template file from the File-Based Data Import for Oracle Supply Chain Management Cloud.
2. Enter data in tabs within the item import 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.

Related Topics
• File-Based Data Import for Oracle Supply Chain Management Cloud guide
• Generate the CSV File: Explained
• Upload to the Universal Content Manager: Explained
• Import Data from the Item Management Interface Tables: Explained

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 Supersede
- 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_TPL_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 in the Setup and Maintenance work area.

### Item Keyword Search Scheduled Process Actions: Explained

The Item Keyword Search scheduled process is used to create and maintain the Item Keyword Search index. The following actions are used for the scheduled process:

- **CREATE**: Creates the index, if not present. This will index all items in the instance.
- **DROP**: Drops the index. You should only use this action if the index is damaged and needs to be indexed from scratch.
- **UPDATE**: This action checks if any additions or deletions have been made to the list of keyword attributes from the Manage Keyword Attributes task. The action will either re-index all rows or the affected rows, depending if seeded attributes have been added or deleted.
- **REBUILD**: This will re-index all rows. You should only use this action if the index is damaged and needs to be rebuilt because it takes a very long time to complete.
- **SYNC**: This will process the index changes in the queue.

You do not need to enter values for any of the remaining fields on the page.

### 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.

FAQS for Defining Items

What are item classes?
The item class hierarchy provides a logical classification and grouping of similar products, and also acts as a template for product definition by enabling the association and inheritance of data elements and policies that are shared by products.

How can I create an item class?
To create an item class, select a parent item class on the Item Class Search Results page and select Create. Provide the required information, and optionally include additional details, such as attribute groups, pages, templates, and search and display formats.

How can I create item class templates?
Create a template for item class and then define the Overview and Specification tabs of the item class template.

What's lifecycle validation?
Lifecycle validation enforces compatible lifecycle phases between parent and component items in an item structure, at the structure name level. For example, an item in the Design lifecycle phase can only have component items that are in the same lifecycle phase or the next phase. The only exception is for items in the Obsolete lifecycle phase, because this is considered the end of life for an item. Items in the Obsolete lifecycle phase cannot be added to a new structure.

The lifecycle phases for a computer component lifecycle might be:

• Concept
• Design
• Prototype
• Production
• Retirement (Obsolete)

What number generation methods are available?
The available number generation methods are: Sequence Generated, User-Defined, and Rules.

Related Topics
• Generating Numbers and Descriptions with Rules: Examples
How can I promote an item's lifecycle phase?

You can change an item’s lifecycle phase based on the lifecycle of the item. Phases associated at the item class will be shown and depending on your business process, can be promoted.

There are business rules associated with these changes. Checks are performed for component’s lifecycle phase and prompt an error message if there are incompatibilities.

How can I demote an item's lifecycle phase?

You change an item’s lifecycle phase based on the life cycle of the item. Phases associated at the item class will be shown and depending on your business process, can be demoted. Changing an item’s lifecycle phase also changes not only its lifecycle phase but also all the components present in the item’s structure, which based on the structure setup.

What's the difference between lifecycle phase types and lifecycle phases?

Lifecycle phase types are seeded and describe the type of lifecycle phase. They are Design, Obsolete, Preproduction or Prototype, and Production.

Lifecycle phases must be created by the user by selecting one of the seeded lifecycle phase types.

What does the All Values Required field, in the Definition Workflow Details table on the Item Management tab of the Edit Item Class task mean?

When checked, all attributes of the associated entity specified on that row must be populated by the assignee in order to for the workflow to proceed to the next step of the definition workflow. For example, if the associated entity is Physical Attributes, then all attributes in the attribute group must be populated by the assignee in the New Item Request workflow step.

Can mandatory definitions be inherited?

If the association is inherited and All Values Required is not checked, then inherited-required attributes from parent will be displayed in read only format under selected list and you can move additional required attributes from available list to selected list.

If the association is inherited and All Values Required is checked in parent, then All Values Required is read only and inherited attributes will be placed under selected list also in read only format.
4 Defining 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 (optional)</td>
<td>Descriptive flexfields can only have one context available at a single time.</td>
</tr>
<tr>
<td>Manage Category Descriptive Flexfields (optional)</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 Oracle Fusion applications, specifically the 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 in the Oracle Fusion Product Hub.
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**

The Detail 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.

Assigning Catalogs and Categories Using Item Rules: Explained

You can define rules that assign items to catalogs and their categories, and rules that validate those assignments.
Catalog and Category Assignment Rules

You can define assignment rules that automatically assign items to one or more catalogs and categories when a condition is satisfied. The condition can be based on attribute values, organization assignments, or other catalog assignments.

The following table summarizes an example of an item rule that:

- Is defined in a rule set that is associated with an item class for items that represent garments.
- Tests whether an item has an attribute Size equal to XS and an attribute Material equal to Wool
- If the result of the test is true, then assign the item to the category Winter in the catalog KidsWear.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Catalog assignment</td>
</tr>
<tr>
<td>Description</td>
<td>Auto assignment of item to catalog/ category based on given attributes.</td>
</tr>
<tr>
<td>Return type</td>
<td>Catalog category</td>
</tr>
<tr>
<td>Primary If Expression</td>
<td>True</td>
</tr>
<tr>
<td>Secondary If Expression</td>
<td>[Item].[Properties].[Size] == &quot;XS&quot;  \ AND \ [Item].[Properties].[Material] == &quot;Wool&quot;</td>
</tr>
<tr>
<td>Return Value</td>
<td>Catalog[KidsWear].Category[Winter]</td>
</tr>
<tr>
<td>User Message</td>
<td>XS, Wool item is assigned to Winter Category under KidsWear Catalog.</td>
</tr>
</tbody>
</table>

Catalog and Category Reassignment Using Rules

Item rules can reassign items to different categories, depending on the scenario.

- If a catalog allows multiple item category assignments, then a reassignment rule creates a new item assignment to a category, even if the item is already assigned to a different category.

If a catalog does not allow multiple item category assignments, then the reassignment rule:

  - Changes the existing category assignment for the item, if an assignment exists
  - Creates a new category assignment for the item, if an assignment does not exist

Example: a rule assigns item A100 to an existing Category Y in Catalog A, where A100 is currently assigned to Category X. Catalog A does not allow multiple item category assignments. The result is:

  - Item A100 is reassigned to from Category X to Category Y.
  - If Catalog A had allowed multiple item category assignments, then item A100 would have been newly assigned to Category Y and would have also remained assigned to Category X.
Catalog and Category Validation Rules
You can define validation rules that validate catalog or category assignments based on attribute values, or on an assignment to other catalogs or an organization assignment.

The following table summarizes an example of an item rule that:

- Is defined in a rule set that is associated with an item class for items that represent garments.
- Tests whether an item is assigned to the category `Summer` in the catalog `LadiesWear`.
- If the result of the test is true, and the item is in the specified category then validate that the item is not also assigned to the category `Summer` in the catalog `Kids`.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Catalog assignment validation</td>
</tr>
<tr>
<td>Description</td>
<td>Validate catalog assignments</td>
</tr>
<tr>
<td>Severity</td>
<td>Reject</td>
</tr>
<tr>
<td>If Expression</td>
<td><code>assignedtoCatalog(Catalog[LadiesWear].Category[Summer])</code></td>
</tr>
<tr>
<td>Validation Expression</td>
<td><code>!assignedtoCatalog(Catalog[Kids].Category[Summer])</code></td>
</tr>
<tr>
<td>User Message</td>
<td>Ladies wear products cannot be assigned to Kids wear catalog.</td>
</tr>
</tbody>
</table>

The function `assignedtoCatalog()` validates whether an item is assigned to the specified catalog or category.

**Related Topics**
- Rules and Rule Sets: Explained
- Item Rule Logical Functions and Operators
- Item Rule Utility Functions

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.

Importing Catalogs with FBDI: Explained
Coming soon in 18C
5 Defining Change Orders

Change Order Setup: Overview

Before you can create change orders, you must complete these tasks in the Setup and Maintenance work area and 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. You can access this task from the Product Management offering and the Change Orders functional 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. You can access this task from the Product Management offering and the Change Orders functional 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 based on configuration in the change order type. The approval can be user defined, rules based, or managed by a web service. 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. You can access this task from the Product Management offering and the Change Orders functional area.

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></td>
<td></td>
<td></td>
<td>Displays the following modifications in redline: item detail, EFF, structure, AML and attachment.</td>
</tr>
<tr>
<td>Change Order without Revision Control</td>
<td>Used when revision control is not required or when changes are minimum.</td>
<td>No</td>
<td>Yes</td>
<td>Yes.</td>
</tr>
<tr>
<td></td>
<td>For example, replacing an existing manufacturer part with a similar part supplied by another manufacturer.</td>
<td></td>
<td></td>
<td>Displays the following modifications in redline: AML, EFF, and item detail.</td>
</tr>
<tr>
<td>Change Request</td>
<td>Used when suggesting changes for a released item.</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>For example, a supplier could request changes in certain specifications.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deviation Change Request</td>
<td>Used when deviating from a process or specification for a specific time period.</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>For example, suggesting the use of a substitute component as a temporary means of resolving an issue.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Commercialization Change Order</td>
<td>Used during the commercialization phase of the product. Can be created and edited in Product Hub. Also available in Product Development as read-only.</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 Setup and Maintenance work area. You can access this task from the Product Management offering and the Product Rules functional area. In this task, select association type as Change order type. On the Workflow tab, edit the change order type to select the entry and exit criteria.
The following attributes may be used to create the validation rules:

- Change Header
  - Priority
  - Reason
  - Need-by Date
  - Requested by
  - Description
  - Descriptive Flexfield

- Change Line
  - Descriptive Flexfield

**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.

**Change Order Approvals: Points to Consider**

For each change order type, you set up the work flow of its approval as either:

- User Defined
- Rules-based

**Change Order Approval Considerations**

You need to consider the following factors when choosing between user defined and rules-based approvals.

- Who needs to perform the approvals. The group of approvers required can vary in many ways: by functional department, by item class, by change priority, and so on.

- Whether approvals need to be performed at header level only, or at both header and line level.
  - In header level approvals, approval at the header level approves all the lines in the change order.
User defined approvers are part of the setup for specific change order types. You associate approvers with a change status, at the header level.

Rules-based approvers are derived by rules defined with Approval Management Extensions (AMX). AMX rules can be set up based on parameters like item classes, change attributes, item attributes, and items. Approvals can be set up at the header level.

User Defined Approvals
User defined approvals are defined in change order types, by assigning a person or role to review change orders of that type. The advantages of user defined approvals include:

- The setups are simpler, and can be maintained by ordinary business users.
- They address most business scenarios.

The disadvantages of user defined approvals include:

- Each approval choice requires its own change order type.

Rules Based Approvals
Rules-based approvals are defined by using Approval Management Extensions. The advantages of rules-based approvals include:

- In complex situations, they can derive different approvers within the same change order type.
- Since a type is not required for each approval choice, fewer change order types are needed.
- Approvals are available at the header level.

Related Topics
- Change Order Approval Process: Explained

Manage Change Order Entry and Exit Rule Sets: Explained

This topic describes how to manage change order entry and exit criteria. Run the following tasks in the Setup and Maintenance work area to manage entry and exit criteria:

- Manage Change Order Entry and Exit Rule Sets.
- Manage Item Rule Sets: Use the Product Management offering and the Product Rules functional area.

You can configure the following field types as required fields when the change order enters or exits a workflow state:

- Change header attributes.
- Descriptive flexfields of the change header and the change line.

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**

- Product Development Components: How They’re Configured
- Setting up Product Development: Roadmap

## Defining Entry and Exit Criteria for Change Orders: Procedure

You can define criteria that govern when a change order 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 a change order type.

Defining entry and exit criteria for a change order type requires the following tasks:

1. Defining the change order type.
2. Defining the criteria in a validation rule set.
3. Assigning the criteria to the change order type’s workflow.

### Defining the Change Order Type

Define a change order type so that its workflow includes one or more status types that can have entry or exit criteria.

You can assign criteria to a change order’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>Interim approval</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Approval</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
Example: Define a change order type so that it contains the workflow status Approval, which can be governed by exit criteria.

1. From the Setup and Maintenance work area, select the Manage Change Order Types task, then create a change order type.
2. On the Workflow tab of the Create Change Order Type page, add the status Approval, after the status Open.
3. Notice that there is a choice list in the Exit Criteria column of the Open status, and also a choice list in the Entry Criteria column of the Approval status. There are no criteria to choose yet. You will create them in the next task.

Defining the Criteria in a Validation Rule Set

Define one or more rule sets that validate your criteria for change 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 change order header rows. The attribute group Change Header Main, which contains attributes corresponding to the basic attributes of a change order 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 change order 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. In the Setup and Maintenance work area, select Product Management offering.
2. Run the Manage Item Rule Sets task.
3. Create a rule set. In the Create Rule Set dialog, you must make the following selections.
   - For Type, select Validations.
   - For Association Type, select Change order type.
   - For Association Name, select the name of the change order type that you defined, which will use the exit criteria that you are about to define.
4. On the Edit Rule Set page, create a rule. The Severity will be automatically set to Reject.
5. 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:
   
\[
\text{[ChangeHeader].[ChangeHeaderMain].[ReasonCode]} \]

6. Add the rule code \( \text{== "Cost"} \), to form the following complete comparison expression:

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

7. In the Validation Condition field, use the same procedure to insert the following expression:

\[
\text{[ChangeHeader].[ChangeHeaderMain].[PriorityCode]} \text{=="High"}
\]

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

Change orders related to cost must have high priority to be considered for approval.
Assigning the Criteria to the Change Order Workflow

Modify a change order type to use a validation criteria rule set.

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

Example: Select the entry criteria for the workflow status Approval, which you added to the change order type that you created.

1. From the Setup and Maintenance work area, search for and select the Manage Change Order Types task, then edit the change order type that you created.
2. On the Workflow tab of the Edit Change Order Type page, select the status Approval, which you previously added.
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 change orders created using this change order type.

Now, when a change order 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

- Change Order Workflow: Explained
- Item Rule Syntax: Explained

Change Order Approval Required Rules: Explained

Rules can require approval through a change order when certain changes are made to an item. Approval required rules are triggered when you make these changes to an item and will add the changes to a new or existing change order.

Approval required rules are supported for the following areas

- Items
- Item revisions
- Item supplier

Changes to item attributes and structures fall under item and item revisions

Changes to item revision attributes and item supplier associations fall under item supplier

If an approval required rule is triggered on any item changes that fall into one of the three areas listed above, then all changes authored for that item that fall under that area will be added to a change order.

For example, a rule is defined that requires approval if a primary structure is created for any item.

During a single session directly editing an item, a user changes the value of an item level attribute and then creates a primary structure for that item.
On saving the data, the approval required rule will be triggered because of the new primary structure and the user is prompted to either create a new change order or add to an existing one.

When the new change order is created or an existing change order is updated, then the change to the value of the item attribute will also get added to the change order along with the primary structure change for required approval.

### Approval Required Rules and User Groups for Approval

The approvals can be from individual users or from multiple users in a predefined user group. Because individual rules can be made for specific item revisions, attributes, or structures one rule can require approvals from one user group when an attribute is changed, while changes made to a different attribute will require approvals from a different user group.

For example, the new primary structure rule described above requires approvals from members of the TestApprovalGroup user group.

Another rule is defined that requires approval if the scheduled date when changes will be implemented for an item is later than the date 01/01/2012, then approvals are required from the COApprovalSAG approval group.

This means that if the user creates a primary structure for that item, and specifies a scheduled date for the item that is later than 01/01/2012, approval responses are required from both the COApprovalSAG and TestApprovalGroup approval groups.

Change order approval rules are defined and managed through BPM Worklist.

### Related Topics
- Change Order Management: Explained

### 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 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.

### Troubleshoot Access to Change Orders: Explained

If the user is unable to view or manage change orders, ensure that the relevant privileges are added to the role.
The table lists the privileges required to view and manage change orders, and the corresponding privilege code.

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Privilege Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approve Item Change Order</td>
<td>EGO_APPROVE_ITEM_CHANGE_ORDER_PRIV</td>
</tr>
<tr>
<td>Generate Item Change Order Report</td>
<td>EGO_GENERATE_ITEM_CHANGE_ORDER_REPORT_PRIV</td>
</tr>
<tr>
<td>Manage Item Change Order</td>
<td>EGO_MANAGE_ITEM_CHANGE_ORDER_PRIV</td>
</tr>
<tr>
<td>Monitor Item Change Order Summary</td>
<td>EGO_MONITOR_ITEM_CHANGEORDER_SUMMARY_PRIV</td>
</tr>
<tr>
<td>Reschedule Item Change Order</td>
<td>EGO_RESCHEDULE_ITEM_CHANGE_ORDER_PRIV</td>
</tr>
<tr>
<td>View Item Change Order</td>
<td>EGO_VIEW_ITEM_CHANGE_ORDER_PRIV</td>
</tr>
</tbody>
</table>

To edit the role, use Security Console.

**Simplified Change Management Interface: Explained**

The Simplified Change Management Interface enables Product Hub users to access change orders through side tabs instead of horizontal tabs. The simplified interface is similar to the change order interface in Product Development.

From the Simplified Change Management Interface, users can:

- Create Commercialization Change Order from master and child organizations. Note that users can only view the other change order types.
- Redline an affected object and view the redline summary.
- Promote or demote the change order through the workflow and initiate notifications for approval.
- Add affected objects belonging to the organization from which change order is created.
- Add affected objects created in Product Development or Product Hub. If the item is created in Product Development, you can only add items in the following lifecycle phases: prototype or preproduction, and production.
- Edit component attribute groups in the item structure of a commercialization change order.
- Modify the effective date in a Scheduled change order. The modified date must be within the start and end date of the updated component in the item structure.

**Restrictions**

In the Simplified Change Management Interface, users cannot:

- Move change lines between change orders.
- Initiate a discussion about change by posting and assigning comments.
Affected Object Versus Change Line

Simplified Change Management Interface and the Product Development work area use "affected object" to indicate an object affected by the change order, whereas Product Hub uses "change line"
Defining 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 can 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 associated with a source of attributes, such as attribute groups, item classes, change types, or structure types. The specific source (such as an attribute group name) is defined as part of the rule set. Rule sets also list valid business entities, such as items, item suppliers, or item revisions. The association of a rule set with a specific source of attributes enables the rule expressions entered in rules in the rule set to be validated by checking for allowable entities.

Each attribute is referenced by its business entity and attribute group name followed by the attribute name. For example, [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.

Keep in mind that:

- If the rule set is associated with an attribute group, then only the attributes in that group can be used in its rules.
- If the rule set is associated with an item class, then only the attribute groups valid for that item class can be used in its rules.

You can set the status of a rule set draft. You can keep a rule set in draft status until the drafting of its rules is complete. If a rule set is in draft status, the rule set is not triggered as regular transactions are completed. During this time, you can run rule impact analysis to simulate and study the impact of the rule sets on a selected set of existing items, enabling you 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 by an asynchronous scheduled process.

The rule sets associated with new item requests and change order types are used to generate identifying numbers for new item requests and change orders.

The types of rule sets and rules are as follows:

- Assignment
- Validation
• Blending
• Composite

Assignment
An assignment rule set determines the value of an item attribute based on the specified condition.

An example of an assignment rule, expressed in ordinary English, 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, in the rule, and is therefore computed first. Generally, rule sets for assignments should be executed before rule sets for validations, so that you can write validation conditions against assignment results that you’re confident are valid.

Validation
A validation rule set validates conditions based on attribute values defined for items. Validation rule sets are typically used to model predefined business rules on items.

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 attributes or extensible flexfields.

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.

The severity of a validation rule determines what action is taken if the validation fails. The severity levels, and their resulting actions, are as follows:

<table>
<thead>
<tr>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warning</td>
<td>The business entity can still be saved. An explanatory message, which you define, is displayed to the end user. The user must acknowledge the warning message to save the item.</td>
</tr>
<tr>
<td>Needs Approval</td>
<td>The attribute data requires that a change order be created for the business entity.</td>
</tr>
<tr>
<td>Reject</td>
<td>The business entity cannot be saved until the attribute validation passes.</td>
</tr>
</tbody>
</table>

The following behavior for validation rules occurs during a runtime end user session:

• Rules are only run for attributes that have changed, except in the following circumstances.
  • The item’s item class has been changed
  • The rule is a valid-component-type rule
  • The rule uses Context attributes, such as Context.ExecutionDateTime
  • The rule uses one of the following functions:
    • exists()
    • loopSum()
- conditionalLoopSum()
- assignedToCatalog()
- assignedToOrg()

- If a validation with a severity of Reject fails, then the entire business entity containing the attribute is rejected.
- If a validation with a severity of Needs Approval fails, then a change order must be submitted and approved.
  - All related attributes also require a change order. In this context, related attributes are those that are used in any validation that uses an attribute that requires a change order. In other words, if any attribute requires a change order, then all the updated attributes in that validation rule (those specified in the **IF Expression** and **Validation Condition** fields of the rule) must also be part of the same change order.
  - If the attributes computed in assignment rules are used in subsequent rules, then they can form a chain of dependencies. In order to ensure that the data remains consistent, the change order requirement is propagated along this dependency chain.
  - The change order requirement propagates only along updated attributes. If an attribute is not updated, then it should not affect other attributes.

**Blending**

When there are multiple spoke systems providing item data, blending rules are applied during import, to control which item attribute values are imported into production from the multiple spoke systems, based on the blending priority for each spoke system that you define in the blending rules.

**Composite**

Composite rule sets can contain both assignment and validation rule sets. Composite rule sets 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 assignment and validation 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**
- Generating Numbers and Descriptions with Rules: Examples

**Defining Rule Sets and Item Rules: Procedures**

You can define item rules to enforce integrity constraints on the attributes of items and structures.

To define item rules, you must:

1. Define rule sets to contain the rules.
   - Rule sets can be used for assignments, validations, or blending.
2. Define rules in the rule sets.
   - The procedure for defining rules differs among assignment, validation, or blending rule sets.
3. Enable the rule sets for execution.

The details for these procedures follow.
Defining Rule Sets

The table following these steps provides details about the fields for defining rule sets.

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, enter a name and optional description.
4. Select a value for Composite.
5. Select a value for Type, to determine the type of the rule set.
6. In the Association Type field, select the type of association to use for determining valid attribute expressions.
7. In the Association Name field, select a name to qualify the association type.
8. Select Draft, so that the rule set will not be triggered until you’re satisfied with its behavior.
9. Click Save and Continue.
10. Create one or more rules within the rule set. Unless you selected Yes for Composite, you can only create rules of the Type that you chose.

Continue to the procedure for creating assignment or validation rules.

<table>
<thead>
<tr>
<th>Rule Set Field</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Name</td>
<td>Visible name for the rule set. You can modify this name later.</td>
</tr>
<tr>
<td>Internal name</td>
<td>Generated automatically when you leave the Display Name field. May contain only letters, digits, underscores, and periods.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional description of the rule set.</td>
</tr>
<tr>
<td>Composite</td>
<td>Values are Yes or No.</td>
</tr>
<tr>
<td></td>
<td>Composite rule sets contain other rule sets, either assignment type, validation type, or both. You can also use composite rule sets to aggregate rule sets that operate on different attribute groups and item classes.</td>
</tr>
<tr>
<td></td>
<td>You don’t select associations for a composite rule set. The associations belong to the rule sets contained in the composite set.</td>
</tr>
<tr>
<td></td>
<td>If you select Yes, then you can select the rule set type as Mixed, so that the composite set can contain a mixture of assignment and validation rule sets.</td>
</tr>
<tr>
<td></td>
<td>Blending rule sets cannot be composite.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of the rule set, which determines the type of rules that can be created in the set. You can’t change the type of a rule set after creating it.</td>
</tr>
<tr>
<td></td>
<td>The available types are:</td>
</tr>
<tr>
<td></td>
<td>• Assignments</td>
</tr>
<tr>
<td></td>
<td>• Validations</td>
</tr>
<tr>
<td></td>
<td>• Blending</td>
</tr>
<tr>
<td></td>
<td>• Mixed</td>
</tr>
</tbody>
</table>
Implementing Product Management

Chapter 6

Defining Product Rules

<table>
<thead>
<tr>
<th>Rule Set Field</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Association Type</td>
<td>Associates the rule set with a set of attributes. When combined with an association name, the association type determines the set of attributes that are considered valid in the rule expressions that you enter. The available association types are:</td>
</tr>
<tr>
<td>Association Type</td>
<td>Associates the rule set with a set of attributes. When combined with an association name, the association type determines the set of attributes that are considered valid in the rule expressions that you enter. The available association types are:</td>
</tr>
<tr>
<td>Association Name</td>
<td>The name of the specific associated type that is the source of valid attributes for rule expressions. For example, if you selected an association type of Attribute group, then search for and select the name of an attribute group.</td>
</tr>
<tr>
<td>Draft</td>
<td>Select to put this rule set into Draft status. Draft status prevents the rule from being triggered during item transactions, and allows rule impact analysis on the rule set.</td>
</tr>
</tbody>
</table>

Note: You can't define a rule independently of a rule set. The rule set provides essential contextual data for the rule.

Defining Item Assignment Rules

Define assignment rules to set the values of item attributes based on conditions that you specify.

1. Create or open a rule set of type Assignments.
2. In the Rules tab of the Edit Rule Set page for the rule set, select Create from the Actions menu.
3. In the Create Rule dialog box, enter an integer in the Sequence field.
4. Provide a name and description for the rule.
5. Select a Return Type.
6. If you selected Return Type as Generic, Target Business Entity is enabled and required.
   a. Select a value for Target Business Entity. Target Attribute Group is then enabled and required.
   b. Select a value for Target Attribute Group. Target Attribute is then enabled and required.
   c. Specify a value for Target Attribute.
7. The following table provides details about the fields defining the header level of the rule. You can modify these values after the rule has been created.

<table>
<thead>
<tr>
<th>Rule Field</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>An integer. This sequence number determines the order in which the rule will be triggered when the rule set is triggered.</td>
</tr>
<tr>
<td>Name</td>
<td>Name for the rule. You can modify this name later.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional description of the rule.</td>
</tr>
</tbody>
</table>
Rule Field | Purpose
--- | ---
Return Type | The result of the assignment defined by the rule. The available return types are:
- **Catalog category**: Assigns an item to a catalog category.
- **Generic**: Assigns a value to an attribute.
- **Item description**: Assigns a description to an item.
- **Item number**: Assigns an item number to an item.
- **Organization**: Assigns an item to an organization.

Target Business Entity | If Return Type is Generic, a business entity must be specified, to provide an attribute association for validating expressions. The entities are:
- **Item**
- **Revision**
- **Supplier**

A rule can assign values to revision attributes or supplier attributes based on conditions on item attributes, but the reverse is not true. That is, a rule cannot include revision attributes or supplier attributes in a condition for a rule whose target attribute is an item attribute.

Target Attribute Group | The attribute group for the target business entity. Only enabled if Return Type is Generic.

Target Attribute | The attribute for the target attribute group. Only enabled if Return Type is Generic.

8. Click **OK** to accept the header fields.
9. In the Details section, enter rule expressions in the **Expression** fields described in the following table.

To ensure that valid attribute group and name expressions are used in rules, you can right-click in one of the expression fields and select **Insert Attribute**, then use the Insert Attribute dialog to populate attribute expressions into the rule fields.

Rule Field | Purpose
--- | ---
Primary If Expression | Optional. An attribute expression that must evaluate to true or false, which determines whether the assignment is to be performed. You can enter true or false as a valid expression. This field can be empty if an expression is not required.

THEN Expression | Required. Composed of:
- **Secondary If Expression**: A condition that further determines whether the assignment is to be performed. This field can be empty if an expression is not required.
- **Return Value**: The value resulting from the assignment. Agrees with the return type that you specified.

You can manage additional rows of the THEN Expression with the edit controls on the task bar. The rows are evaluated in the sequence of the expression rows. The execution is halted when the first THEN expression evaluates to True.
Implementing Product Management

Chapter 6
Defining Product Rules

Oracle SCM Cloud

Defining Item Validation Rules
Define validation rules to validate the values of item attributes based on conditions that you specify.

1. Create or open a rule set of type **Validations**.
2. In the **Rules** tab of the Edit Rule Set page for the rule set, select **Create** from the Actions menu.
3. In the Create Rule dialog box, enter an integer in the **Sequence** field.
4. Provide a name and description for the rule.
5. If you select **Valid Component Rules**, then
6. Select a level of **Severity** for the action produced by the rule if the validation fails. Severity only applies to validation rules.
   Select **Valid Component Rules** only if your rule is validating component types, using the `component_type()` function.
7. The following table provides details about the fields defining the header level of the rule. You can modify these values after the rule has been created.

<table>
<thead>
<tr>
<th>Rule Field</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sequence</strong></td>
<td>An integer. This sequence number determines the order in which the rule will be triggered when the rule set is triggered.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Name for the rule. You can modify this name later.</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Optional description of the rule.</td>
</tr>
<tr>
<td><strong>Severity</strong></td>
<td>The severity of the action that is produced if the validation fails.</td>
</tr>
<tr>
<td></td>
<td>o <strong>Warning</strong>: The explanation message is displayed to the user, but the business entity can still be saved</td>
</tr>
<tr>
<td></td>
<td>o <strong>Needs Approval</strong>: The data requires that a change order be created.</td>
</tr>
<tr>
<td></td>
<td>o <strong>Reject</strong>: The business entity cannot be saved until the validation passes. You can increase the severity of this validation result by selecting <strong>Stop further processing when rejected</strong>.</td>
</tr>
</tbody>
</table>

8. Click **OK** to accept the header fields.
9. In the Details section, enter rule expressions in the **Expression** fields described in the following table.

To ensure that valid attribute group and name expressions are used in rules, you can right-click in one of the expression fields and select **Insert Attribute**, then use the Insert Attribute dialog to populate attribute expressions into the rule fields.

<table>
<thead>
<tr>
<th>Rule Field</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>If Expression</td>
<td>Optional. An attribute expression that must evaluate to true or false, which determines whether the validation is to be performed. The expression must evaluate to true for the validation to occur. You can enter true or false as a valid expression. This field can be empty if an expression is not required.</td>
</tr>
<tr>
<td>Validation Condition</td>
<td>Required. This condition is evaluated when the result of the If Expression is true. When the validation condition evaluates to false, then the validation fails, and the action for the selected severity is executed. The User Message is also displayed, for failures.</td>
</tr>
<tr>
<td>User Message</td>
<td>Optional. Displayed to the end user only when the <strong>Validation Condition</strong> evaluates to <strong>false</strong>.</td>
</tr>
</tbody>
</table>

It’s not necessary to check whether attribute values are null (which includes division by zero). Null values normally cause the validation to be ignored, unless you explicitly want to test for a null value with the `isnull()` function. This feature facilitates incremental processing of an item, where the item might not be completely filled in until the end of a process.

10. Click **Validate** to validate the expressions in the rule. A message informs you whether there is an error, or confirms that the rule definition is valid.

11. Define more rules in the rule set, as needed. You can modify the execution order of the rules by editing their sequence numbers.

12. Click **Save** to save the rule set.

13. Add the rule set to the master rule set, to enable its operation during item transactions.

### Enabling Rule Sets and Rules

Before a rule can run, you must enable the rule set that contains it. To enable a rule set, you must assign it to the master rule set, either directly, or by adding it to a rule set that’s included in the master rule set.

The master rule set (named `MASTER_RULE_SET`) is a predefined rule set that contains other rule sets, which can in turn include other rule sets. In each composite rule set, a sequence number controls the order in which its included rule sets are to be executed at run time.

1. On the Manage Rule Sets page, search for `MASTER_RULE_SET` and edit it.
2. On the Edit Rule Set page for `MASTER_RULE_SET`, on the Included Rule Sets tab, select **Add** from the Actions menu.
3. In the Select and Add: Rule Sets dialog box, search for your rule set by name, or by one of the other fields that define a rule set.
4. Select your rule set from the search results, and click **OK**.
5. Your rule set appears in the list of rule sets included in the master rule set, with the next available sequence number, which determines the order of execution of the included rule set. You can select **Edit Sequence** from the Actions menu to modify the sequence.
6. Click **Save** to save the master rule set.
7. On the Edit Rule Set page for your own rule set, the Rule Sets Where Included tab shows that your rule set is included in the master rule set.
Creating an Item Rule Set and Rules: Worked Example

This example demonstrates how to create rule sets and rules to implement integrity constraints for a computer motherboard. The following table summarizes the attribute value assignments required for this scenario.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lead Percent</td>
<td>Is Total Lead Mass divided by Unit Weight</td>
</tr>
<tr>
<td>Sellable Date (date when item can be sold)</td>
<td>Is 10 days after the Availability Date</td>
</tr>
</tbody>
</table>

The following table summarizes the attribute value validations required for this scenario.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum CPU Speed (in MHz)</td>
<td>Must be less than Maximum CPU Speed</td>
</tr>
<tr>
<td>Purchasable</td>
<td>If set to Yes, then a value for List Price must be set</td>
</tr>
<tr>
<td>Unit Height</td>
<td>Cannot change by more than 3% without requiring approval</td>
</tr>
</tbody>
</table>

Task Summary

To implement the above constraints, complete the following tasks:

1. Creating an assignment rule set.
2. Creating assignment rules.
3. Creating a validation rule set.
5. Creating a composite rule set.
6. Adding the composite rule set to the master rule set.

Creating an Assignment Rule Set

In general, rule sets that perform assignments should be executed before validations, so that validation rules can ensure that the results of the assignments are valid.

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, enter a name and optional description. The internal name will not be changeable after the rule set is created.
4. For **Composite**, select **No**.
5. For **Type**, select **Assignments**.
6. For **Association Type**, select **Item Class**.
7. For **Association Name**, select **Chemicals**.
8. Click **Save and Continue**.

The Edit Rule Set page shows a rule set with the specified attribute context, but no rules yet. All the attribute groups in the selected item class are available for us in item rules.

### Creating Assignment Rules

The assignment to be made by the first rule is:

- **Lead Percent** is **Total Lead Mass** divided by **Unit Weight**

1. In the **Rules** tab of the Edit Rule Set page, select **Create** from the Actions menu.
2. In the Create Rule dialog box, enter or select the values shown in the following table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>10</td>
</tr>
<tr>
<td>Name</td>
<td>Lead Percent</td>
</tr>
<tr>
<td>Description</td>
<td>Lead Percent is Total Lead Mass divided by Unit Weight</td>
</tr>
<tr>
<td>Return Type</td>
<td>Generic</td>
</tr>
<tr>
<td>Target Business Entity</td>
<td>Item</td>
</tr>
<tr>
<td>Target Attribute Group</td>
<td>Hazard (for an example user-defined attribute group (EFF))</td>
</tr>
<tr>
<td>Target Attribute</td>
<td>Lead Percent (for an example user-defined attribute group (EFF))</td>
</tr>
</tbody>
</table>

3. Click **OK** to accept the header fields.
4. In the Details section, enter rule expressions in the expression fields described in the following table.

To ensure that valid attribute group and name expressions are used in rules, you can right-click in one of the expression fields and select **Insert Attribute**, then use the Insert Attribute dialog to populate attribute expressions into the rule fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary If Expression</td>
<td>(You can leave this field empty. The value true is inserted when this field is null.)</td>
</tr>
<tr>
<td>THEN Expression</td>
<td>(Consists of the Secondary If Expression and Return Value shown in the following rows)</td>
</tr>
<tr>
<td>Secondary If Expression</td>
<td>(You can leave this field empty. The value true is inserted when this field is null.)</td>
</tr>
</tbody>
</table>
Field | Value
--- | ---
Return Value | \( \text{round}([(\text{Item}.[\text{Hazard}].[\text{Lead Mass}]) / (\text{Item}.[\text{Physical Attributes}].[\text{Unit Weight}]) \times 100, 2]) \)
User Message | Enclosing an attribute expression with dollar sign symbols (\$) renders the current value of the attribute in the displayed user message. For the example shown here, the message is:

5. Click **Validate** to validate the expressions in the rule.
6. Click **Save** to save the rule set.

Saving the rule set enters the first assignment rule into set. Since rules are executed in the sequence you specify, you can ensure that values needed by attributes in later expressions are calculated in earlier expressions.

The assignment to be made by the second rule is:

- Sellable Date Is 10 days after the Availability Date

This assignment is equivalent to the code in the following example, which shows how you can use conditional processing in an item rule:

```plaintext
if ([Item].[Main].[Sellable Flag] == "Yes")
    if ([Item].[Item basic].[Item Class] == "Perishables")
        [Item].[Marketing].[Sellable Date] = [Item].[Planning].[Availability Date] + 3
    else if ([Item].[Item basic].[Item Class] == "Consumables")
        [Item].[Marketing].[Sellable Date] = [Item].[Planning].[Availability Date] + 6
    else
        [Item].[Marketing].[Sellable Date] = [Item].[Planning].[Availability Date] + 10
```

The following steps show how to express that code block in an item rule definition.

7. In the **Rules** tab of the Edit Rule Set page, select **Create** from the Actions menu.
8. In the Create Rule dialog box, enter or select the values shown in the following table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>20</td>
</tr>
<tr>
<td>Name</td>
<td>Sellable Date Calculation</td>
</tr>
<tr>
<td>Description</td>
<td>Calculate Sellable date based on the Availability Date.</td>
</tr>
<tr>
<td>Return Type</td>
<td>Generic</td>
</tr>
<tr>
<td>Target Business Entity</td>
<td>Item</td>
</tr>
<tr>
<td>Target Attribute Group</td>
<td>Marketing</td>
</tr>
<tr>
<td>Target Attribute</td>
<td>Sellable Date</td>
</tr>
</tbody>
</table>

9. Click **OK** to accept the header fields.
10. In the Details section, enter rule expressions in the expression fields described in the following table. You will have to add an additional two rows of **Then Expression** by selecting the last row in the table and clicking **Add Row**.
### Defining Product Rules

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary If Expression</td>
<td>[Item].[Main].[Sellable Flag] == &quot;Yes&quot;</td>
</tr>
<tr>
<td>(first) THEN Expression</td>
<td>(Consists of the Secondary If Expression and Return Value shown in the following rows)</td>
</tr>
<tr>
<td>Secondary If Expression</td>
<td>[Item].[Item basic].[Item Class] == &quot;Perishables&quot;)</td>
</tr>
<tr>
<td>Return Value</td>
<td>[Item].[Planning].[Availability Date] + 3</td>
</tr>
<tr>
<td>(second) THEN Expression</td>
<td></td>
</tr>
<tr>
<td>Secondary If Expression</td>
<td>[Item].[Item basic].[Item Class] == &quot;Consumables&quot;</td>
</tr>
<tr>
<td>Return Value</td>
<td>[Item].[Planning].[Availability Date] + 6</td>
</tr>
<tr>
<td>(third) THEN Expression</td>
<td></td>
</tr>
<tr>
<td>Secondary If Expression</td>
<td>(This field remains empty.)</td>
</tr>
<tr>
<td>Return Value</td>
<td>[Item].[Planning].[Availability Date] + 10</td>
</tr>
<tr>
<td>User Message</td>
<td>(This field remains empty.)</td>
</tr>
</tbody>
</table>

Leaving the last **Secondary If Expression** blank means it defaults to `true` and is always selected. No **User Message** was entered, so nothing is shown to the user.

11. Click **Validate** to validate the expressions in the rule. Click **Save** to save the rule set. The assignment rule set is now complete.

## Creating a Validation Rule Set

Validation rule sets are created in much the same way as assignment rule sets, except that the **Type** is set to **Validation**.

1. On the Manage Rule Sets task page, select **Create** from the task bar.
2. In the Create Rule Set dialog box, enter a name and optional description.
3. For **Composite**, select **No**.
4. For **Type**, select **Validations**.
5. For **Association Type**, select **Item Class**.
6. For **Association Name**, select **Computers** or **Laptops**.
7. Click **Save and Continue**.
Creating Validation Rules

The validation to be made by the first rule is:

- Minimum CPU Speed (in MHz) Must be less than Maximum CPU Speed

1. In the Rules tab of the Edit Rule Set page, select Create from the Actions menu.
2. In the Create Rule dialog box, enter or select the values shown in the following table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>10</td>
</tr>
<tr>
<td>Name</td>
<td>Maximum Minimum CPU Check</td>
</tr>
<tr>
<td>Description</td>
<td>Check that Minimum CPU Clock Speed is less than Maximum CPU Clock Speed.</td>
</tr>
<tr>
<td>Severity</td>
<td>Reject</td>
</tr>
</tbody>
</table>

3. Click OK to accept the header fields.
4. In the Details section, enter rule expressions in the expression fields described in the following table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF Expression</td>
<td>(This field remains empty.)</td>
</tr>
<tr>
<td>Validation Condition</td>
<td>[Item].[Motherboard Spec].[Min CPU Speed] &lt;= [Item].[Motherboard Spec].[Max CPU Speed]</td>
</tr>
<tr>
<td>User Message</td>
<td>Minimum CPU Speed has to be less than or equal to Maximum CPU Speed.</td>
</tr>
</tbody>
</table>

5. Click Validate to validate the expressions in the rule. Click Save to save the rule set.

The validation to be made by the second rule is:

- If Purchasable is set to Yes, then a value for List Price must be set

This validation demonstrates how the value of one field must be value based on the value of another field.

6. In the Rules tab of the Edit Rule Set page, select Create from the Actions menu. In the Create Rule dialog box, enter or select the values shown in the following table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>20</td>
</tr>
<tr>
<td>Name</td>
<td>Require List Price</td>
</tr>
<tr>
<td>Description</td>
<td>List Price must be set if Purchasable is set to Yes</td>
</tr>
</tbody>
</table>
7. Click **OK** to accept the header fields. In the Details section, enter rule expressions in the expression fields described in the following table. The `!` in the function expression `!isnull()` negates the logical result of the expression.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field ID</td>
<td>Value</td>
</tr>
<tr>
<td>Severity</td>
<td>Reject</td>
</tr>
</tbody>
</table>

8. Click **Validate** to validate the expressions in the rule. Click **Save** to save the rule set.

The validation to be made by the third rule is:
- Unit Height cannot change by more than 3% without requiring approval.

This validation demonstrates how to check the new attribute value against the value that is currently in production. If this validation fails, then the user will be required to create a change order for this change.

9. In the **Rules** tab of the Edit Rule Set page, select **Create** from the Actions menu. In the Create Rule dialog box, enter or select the values shown in the following table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>30</td>
</tr>
<tr>
<td>Name</td>
<td>Height Change</td>
</tr>
<tr>
<td>Description</td>
<td>Unit Height cannot change by more than 3%.</td>
</tr>
<tr>
<td>Severity</td>
<td>Needs Approval</td>
</tr>
</tbody>
</table>

10. Click **OK** to accept the header fields. In the Details section, enter rule expressions in the expression fields described in the following table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF Expression</td>
<td>(This field remains empty.)</td>
</tr>
<tr>
<td>Validation Condition</td>
<td><code>abs(percent([Item].[Physical Attributes].[Height])) &gt; 3.0</code></td>
</tr>
<tr>
<td>User Message</td>
<td>A unit height change of 3% requires a change order.</td>
</tr>
</tbody>
</table>
The `percent()` function returns the percentage change of the new value to the value in production. This change can be positive or negative, hence the `abs()` function converts it to an absolute change.

11. Click **Validate** to validate the expressions in the rule. Click **Save** to save the rule set.

The validation rule set is now complete. Click on the **Usage** tab to review all the attributes used in this rule set. Click **Save and Close** to return to the Manage Rule Sets page.

---

### Creating a Composite Rule Set

Now create a composite rule set that includes both of the rule sets that you just created. This is the rule set that you will assign to the Item entity in the final step to activate the rules we have created.

1. On the Manage Rule Sets task page, select **Create** from the task bar.
2. In the Create Rule Set dialog box, enter a name and optional description.
3. For **Composite**, select **Yes**.
4. For **Type**, select **Mixed**.

In more complex scenarios, you might have multiple layers of composite rule sets, some of which you only want to contain validations or assignments rules. But eventually you will want to include both kinds of rules.

5. Click **Save and Continue**.
6. On the Included Rule Sets tab of the Edit Rule Set page for the new composite rule set, select **Add** from the **Actions** menu.
7. In the Select and Add: Rule Sets dialog box, search for and select the assignment rule set that you created, then click the **Apply** button.
8. Repeat the search and selection step to add the new validation rule set to the composite rule set.
9. The rule sets will be executed in the order that they’re listed in the Included Rule Sets list, so it’s important to put validations after assignments if you want to validate some of the assignment results that you calculated.
10. Click **Save and Close**.

---

### Adding the Composite Rule Set To the Master Rule Set

The rules you created are not active until you add them to the Master Rule Set, as described in the following steps. Also, note that the **Draft** check box should be deselected for a rule set to be executed at runtime. Draft rule sets are not run during regular transactions.

1. On the Manage Rule Sets page, search for MASTER_RULE_SET and edit it.
2. On the Edit Rule Set page for MASTER_RULE_SET, on the Included Rule Sets tab, select **Add** from the **Actions** menu.
3. In the Select and Add: Rule Sets dialog box, search for your composite rule set by name, or by one of the other fields that define a rule set.
4. Select your rule set from the search results, and click **OK**.
5. Your rule set appears in the list of rule sets included in the master rule set, with the next available sequence number, which determines the order of execution of the included rule set. You can select **Edit Sequence** from the Actions menu to modify the sequence.
6. Click **Save** to save the master rule set.

You are now ready to test the validations. When you make the relevant item updates, the updated values should be validated by the rules that are triggered, and the error messages should appear.
You can also examine some of the other tabs on the Edit Rule Set page. In the validation and assignment rule sets the Rule Sets Where Included tab should now show the composite rule set that you added them to. In the composite rule set, the Attributes section on the Usages tab should show the item assignments for the rule sets.

**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.

<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:

\[ \text{[Item].[Logistics].[StartDate]} > "2005-Feb-23" \]

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.

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:

\[ \text{[Item].[Logistics].[SellDate]} > \text{[Logistics].[AvailableDate]} + 30 \]

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:

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

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:

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

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 (\( . \)). You can use the display name of an attribute naming element, which can include space characters, but must be enclosed in single brackets.

Following is the syntax of an attribute expression:

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

When referencing descriptive flexfields, use the segment code, as in the following example:

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

### UOM Expressions

If an attribute definition includes a unit of measure (UOM), you can access the unit by appending `UOM` to the attribute expression.
The following example shows the syntax for an expression that accesses the UOM for an attribute:

```
[Item].[<Attribute Group Name>].[<Attribute> Unit of Measure]
```

For example, the following expression might return \( \text{kg} \) if the attribute’s UOM is kilograms:

```
[Item].[Physical Attributes].[Weight Unit of Measure]
```

So for this attribute you can use expressions like:

```
[Item].[Physical Attributes].[Weight Unit of Measure] == "K"
```

All comparisons between amounts are automatically adjusted to account for different UOMs. Accessing the UOM of an attribute explicitly should not be necessary for comparison purposes.

### 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:

```
if (not isnull(Item.PhysicalAttributes.Weight)) then Item.PhysicalAttributes.Weight <= 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 `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 **If Expression** and **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 \texttt{expression1} and \texttt{expression2}. Returns null if any argument is null.

\texttt{expression1 - expression2}

Performs regular mathematical subtraction of \texttt{expression2} from \texttt{expression1}. Returns null if any argument is null.

\texttt{expression1 * expression2}

Performs regular mathematical multiplication of \texttt{expression1} and \texttt{expression2}. Returns null if any argument is null.

\texttt{expression1 / expression2}

Performs regular mathematical division of \texttt{expression1} by \texttt{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 \texttt{expression1} and \texttt{expression2}.

\texttt{sum(expression1, expression2, expression3, \ldots)}

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:

\texttt{abs(expression)}

Returns the absolute value of \texttt{expression}.

Example:

In this example, the percentage weight change has to be less than 10.

\texttt{abs(percent([Item].[Physical Attributes].[Unit Weight])) <= 10}

**ConvertToUOM**

Syntax:

\texttt{ConvertToUOM(expression, "target UOM")}

Returns an amount in the specified \texttt{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.

\texttt{ConvertToUOM([Item].[Physical Attributes].[Unit Weight], "kg") <= 10}

**max**

Syntax:

\texttt{max(expression1, expression2, \ldots)}
Returns the maximum value of a series of values. Can also be used for arrays or in query expressions.

Example:
In this example, the result would be 2006-11-30.

\[
\text{max}("2006-10-12", "2006-11-30")
\]

\textbf{min}

Syntax:

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

Returns the minimum value of a series of values. Can also be used for arrays or in query expressions.

Example:
In this example, the result would be 2006-10-12.

\[
\text{min}("2006-10-12", "2006-11-30")
\]

\textbf{round}

Syntax:

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

Rounds a number to the specified number of decimal places. round rounds to the nearest value.

Example:
In this example, the returned value is 1.58.

\[
\text{round}(1.5758, 2)
\]

\textbf{rounddown}

Syntax:

\[
\text{rounddown(expression, decimal_places)}
\]

Rounds a number to the specified number of decimal places. rounddown rounds toward zero.

Example:
In this example, the returned value is 1.57.

\[
\text{rounddown}(1.5758, 2)
\]

\textbf{roundup}

Syntax:

\[
\text{roundup(expression, decimal_places)}
\]

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.

\[
\text{roundup}(1.5758, 2)
\]
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.

<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</td>
<td>null</td>
<td>null</td>
</tr>
</tbody>
</table>

percent

Syntax:
- `percent(attribute)`
Returns the difference between new and current production values, expressed as a percentage, according to the following formula:

\[
\frac{\text{delta}(\text{attribute})}{\text{current_production_value}} \times 100
\]

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:

```
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:

```
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.

toLowerCase

Syntax:

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

**substring**

Syntax:

```plaintext
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 in a validation rule to perform this kind of validation.

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

**trim**

Syntax:

```plaintext
trim(expression)
```

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

**uppercase**

Syntax:

```plaintext
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>expression1</th>
<th>expression2</th>
<th>expression1 and expression2</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>*[see note]</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 Extra Data Servers, the organization cannot be S2.

Severity: Reject
If Expression: `[Item].[Item Basic].[Item Class] == "Extra Data 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])`
component_type

Syntax:

    component_type("typename1", "typename2", ...)

In a validation expression that checks the component types for a structure, returns TRUE if the components associated with the structure are valid. The argument is a list of valid user item types for the components.

When creating a rule that checks the component types for a structure, you must select the Valid Component Rules check box in the Create Rules dialog box.

Unlike other validation rules, rules that validate components do not have a Severity.

Example:

In the following example, when the structure is Primary, only type1 and type2 can be used as components of the structure.

    If Expression: [Structure].[Structure Attributes].[Structure Name] == "Primary"
    Validation Condition: component_type("type1","type2")

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 evaluates 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].[GTIN Main].[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.

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

**from_item_class**

Syntax

\begin{verbatim}
from_item_class ("item class name")
\end{verbatim}

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 "item class name", which must be enclosed in quotation marks.

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

\begin{verbatim}
FROM_ITEM_CLASS("TCParent")
\end{verbatim}

Combined example:
The following example prevents changing the item class of an item from Pneumatic_Pumps to Hydraulic_Pumps.

\begin{verbatim}
Severity: Reject
If Expression:
FROM_ITEM_CLASS("Pneumatic_Pumps") && TO_ITEM_CLASS("Hydraulic_Pumps")
\end{verbatim}

**to_item_class**

Syntax

\begin{verbatim}
to_item_class("item class name")
\end{verbatim}

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 "item class name", which must be enclosed in quotation marks.

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

\begin{verbatim}
TO_ITEM_CLASS("TCCHLDVR")
\end{verbatim}

**Descriptive Flexfields**

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

\begin{verbatim}
[<Entity Name>].Flexfield[<Flexfield segment code>]
\end{verbatim}

Example:

\begin{verbatim}
[ChangeHeader].Flexfield[Product__Line]
\end{verbatim}

**Comparison Operators**

The comparison operators available in item rules are listed below.

- \texttt{==} (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:

```
expression1 == expression2  (equals
expression1 != expression2
expression1 < expression2
expression1 <= expression2
expression1 > expression2comp
expression1 >= expression2
```

`isnull`

Syntax:

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

**Item Rule Expressions with Business Entities: Explained**

Item rules refer to attributes through references that include the business entity that includes the attribute.

**Supported Business Entities**

When you reference an attribute in rule expressions, specify the business entity as the first term in the expression, as shown in the following examples:

Syntax:

```
[entity name].[attribute group name].[attribute]
```
Example:

\[\text{[Item].[Main].[Item Status]}\]

The business entities supported for use in item rules are listed in the following table:

<table>
<thead>
<tr>
<th>Entity</th>
<th>For assignment rules</th>
<th>For validation rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross Reference</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>GTIN</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Item</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Related Item</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Revision</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Structure</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Style Item</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Supplier</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Cross-Entity References

You can reference attributes from certain other entities while creating rules for an attribute belonging to an entity. For instance, you can create a validation constraint for a supplier-level attribute that references an item-level attribute, as summarized in the following example:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severity</td>
<td>Reject</td>
</tr>
<tr>
<td>IF Expression</td>
<td>[Supplier].[Promotion Attributes].[Discount] &gt;= 20%</td>
</tr>
<tr>
<td>Validation Condition</td>
<td>[Item].[Price Attributes].[MSRP] &gt; 10$</td>
</tr>
</tbody>
</table>

The following example summarizes another validation rule that references attributes from different entities, namely the item and cross-reference relationship entities:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severity</td>
<td>Warning</td>
</tr>
<tr>
<td>IF Expression</td>
<td>[Item].[Item_Basic].[ITEM_CLASS] == &quot;Electronics&quot;</td>
</tr>
</tbody>
</table>
Using GTIN Entities

You can reference GTIN (Global Trade Identification Number) attributes to enforce your own business validations with regard to GTIN association.

Based on the specified criteria involving attributes or organizations, GTIN validation rules can check whether a GTIN is associated to an item, or check whether there was any change in the associated GTIN.

GTIN rules can also validate the digits of an item's GTIN. You can use a rule 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 "Case", then the fourth digit of the GTIN should be 8. The following example summarizes a validation rule that references attributes for Item and GTIN entities.

Using Style Item Entities

You can use the Style Item entity to access the attributes of the corresponding style item of SKU items.

If a rule contains a Style Item entity then the rule will only be run if:

- The user modified a style item.
- The user created or modified a SKU item.
- The user created or modified an entity (such as a revision, or supplier association) on a SKU item and the rule also uses that entity.

The following example summarizes an assignment rule that references attributes from the Style Item entity:
Field | Value
--- | ---
Return Value | 
| [StyleItem].[Main].[Item Description] + ", " + [Item].[Top Variants].[Color] + ", " + [Item].[Top Variants].[Top Size]

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])
```

Compares `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:

```
in(expression, value1, value2,...)
```

Returns TRUE if `expression` is found in `value1`, `value2`, or other following value arguments. Returns FALSE otherwise. Requires specifying at least `expression` and `value1`. You can specify an unlimited number of `value` arguments.

Example:

The following example returns TRUE if "RED" is contained in the value of either of the specified attributes for colors.

```
If Expression: in("RED", [Item].[BODY ATTR].[COLOR], [Item].[COVER ATTR].[COLOR])
```

Note: The functions `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:

```
ot_in(expression, value1, value2,...)
```

Returns TRUE if `expression` is not found in `value1`, `value2`, or other following value arguments. Returns FALSE otherwise. Requires specifying at least `expression` and `value1`. You can specify an unlimited number of `value` arguments.

Example:

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

```
If Expression: not_in("RED", [Item].[BODY ATTR].[COLOR], [Item].[COVER ATTR].[COLOR])
```

to_number

Syntax:

```
to_number(string)
```

Returns `string` as a value of type Number.

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

```
to_number([Item].[PHYSICAL PROP].[COUNT])
```

### Custom Function Calls in Web Services

You can use custom functions in rule expressions by first writing functions and creating a public web service that includes the custom functions, then using `InvokeWebService()` to call the custom functions.

#### InvokeWebService

**Syntax:**

```
InvokeWebService("service_name", "function_name", input_argument1, input_argument2,...)
```

The names of the web service and its custom function are passed as quoted strings which cannot include space characters. Item rules support only simple type parameters and not complex type parameters being passed to this function.

Returns a value produced by a custom function called through a web service. The web service function must return a single value, since this function does not accept more than one value.

**Example:**

```
InvokeWebService("ValidateSLN_Digit", "execute", [Item].[Tag AG].[SL No])
```

When a rule expression using `InvokeWebService()` is executed at runtime, `InvokeWebService()` calls the web service `service_name`, and passes the input payloads such as `input_argument1` to the function `function_name`. The the value returned from `function_name` is returned to the rule expression by `InvokeWebService()`.

Before calling the function `function_name` in a rule expression, you must register the web service `service_name`, along with the credentials used to call the service.

### 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:**

```
expression1 + expression2
expression1 - expression2
```

**Example:**

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

```
[Item].[Logistics].[LeadTime] + 3
```

### Attributes Supported for Item Rules

All user-defined attributes (extensible flexfields) can be accessed using the syntax `[entity name].[attribute group name].[attribute name]`. For example: `[Item].[Physical Attributes].[Unit Weight]`. In addition, rules support access to the predefined attributes listed here, using predefined attribute groups to access the attributes.
Rule Set Context Attributes

This is a special set of attributes that refers to the context of a rule’s execution. Expressions using these attributes are formed with the prefix `Context` in place of an attribute group name.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleSetVersion</td>
<td>Currently always set to 1.0.</td>
</tr>
<tr>
<td>ExecutionDate</td>
<td>Date for which rules are invoked.</td>
</tr>
<tr>
<td>ExecutionDateTime</td>
<td>Date and Time for which rules are invoked.</td>
</tr>
<tr>
<td>BatchID</td>
<td>Set to Batch ID if running Import.</td>
</tr>
<tr>
<td>BatchName</td>
<td>Set to Batch Name if running Import.</td>
</tr>
</tbody>
</table>

Example:

```
Context.ExecutionDate >= '7/6/2007'
```

Item Primary Attributes

Attribute group name: Main

For the attributes in this group, see the information about item specifications and attributes in the Using Product Master Data Management guide. This group includes the following attributes:

- Approval Status
- Conversions
- Defaulting Control
- Formatted Description
- Item Description
- Item Status
- Lifecycle Phase
- Long Description
- Negative Deviation Factor
- Pack Type
- Positive Deviation Factor
- Pricing
- Primary Unit of Measure
- Secondary Unit of Measure
- Style Item
- Tracking Unit of Measure
- User Item Type
Example:

```
[Item].[Main].[Primary Unit of Measure] == 'Dozen'
```

**Item Basic Attributes**

Attribute group name: Item Basic

For the attributes in this group, see the information about item specifications and attributes in the Using Product Master Data Management guide. This group includes the following attributes:

- Item
- Item Class
- Organization Code
- Style Item
- Approval Status

Example:

```
[Item].[Item Basic].[Approval Status] == 'Approved'
```

**Structure Attributes**

Attribute group name: Structure Attributes

Structure attributes are accessed only in the context of the business entity Item, using the `StructureAttributes` attribute group name.

For the attributes in this group, see the information about item specifications and attributes in the Using Product Master Data Management guide. This group includes the following attributes:

- Structure Name
- Organization Code
- Common Item Name
- Common Organization Code
- Common Structure Name

Example:

```
[Structure].[Structure Attributes].[Structure Name] == 'Primary'
```

**Item Supplier Association Attributes**

Attribute group name: Intersection Primary

In the context of the Supplier business entity, you can access item supplier attributes, intersection attributes (Primary indicator and association Status) and other supplier and supplier address related attributes.

The following example checks whether the item supplier association status is Active.

```
[Supplier].[Intersection Primary].[Status] == 1
```

**Supplier Attributes**

Attribute group name: Supplier
This group includes the following attributes:

- Supplier Name
- Supplier Number
- D-U-N-S
- Taxpayer ID
- Tax Registration Number

Example:

```
[Supplier].[Item Supplier Site Organization].[Supplier Name] = 'Acme'
```

**Item Relationship Attribute Groups: Cross Reference Attributes**

Attribute Group Name: Cross Reference Main

This group includes the following attributes:

- Type
- Description
- Value
- ApplicableOrganization

Example:

```
[CrossReference].[Cross Reference Main].[Type] = 'Old Item Number'
```

**Item Relationship Attribute Groups: Related Item Attributes**

Attribute Group Name: Related Item Main

This group includes the following attributes:

- Type
- Description
- Start Date
- End Date
- Reciprocal
- Planning Enabled

Example:

```
[RelatedItem].[Related Item Main].[Type] = 'Cross-Sell'
```

**Related Topics**

- Item Specifications and Attributes: Explained
- Oracle SCM Cloud Using Product Master Data Management
Rule Impact Analysis: Explained

You can use rule impact analysis to assess the effect of item rules on a given set of existing items.

Purpose of Impact Analysis

When you add new item rules, or modify existing rules, it's important to know the impact the rules have on production item data.

By running a scheduled process for rule impact analysis, you can perform a what-if analysis of the effect that draft rule sets would have on existing items. After the process runs, you examine the results of the analysis. If the results are acceptable, you can add the impacted items to a batch, and import the batch to update the items. For convenience, you can select items by defined criteria (rather than individually), schedule the analysis process (rather than running it manually), and automatically run data updates after analysis (rather than submitting them separately).

Defining an Impact Analysis: Procedure

Impact analysis is based on the ability to designate item rule sets as being in Draft status, and on defining the scope of a particular analysis by selecting a set of items on which to analyze results.

To define an impact analysis:

1. In the Product Information Management work area, select the Manage Item Rule Sets task.
2. Create a new rule set, select the Draft check box and save the rule set. This is the rule whose impact you want to analyze before putting it into production. You can also edit an existing rule set to put it in draft status.
3. Repeat this step for other rule sets to analyze.

Note: Draft rule sets do not operate on production data during ordinary transactions.

4. To enable the running of your draft rule sets, add them to the master rule set, which is named MASTER_RULE_SET. In the master rule set, specify the sequence number that determines the order of execution of the included rule sets.
5. In the Product Information Management work area, select the Analyze Item Rule Set Impact task.
6. The Analyze Item Rule Set Impact page contains a list of all of the rule impact analyses. For each analysis, key information is displayed:
   - The name of the analysis, as a link. Click the link to access the analysis details.
   - Status: whether the analysis has been Completed, or Not Started.
   - If Completed, the date the analysis was submitted.
   - A graphic link showing the number of items impacted, or not, by the rule sets in the analysis. Click the link to access the analysis details.
   - Autoupdate: Whether rules that are not in draft status will automatically update items after the analysis runs.
   - Import Status: Whether the import of the impacted items has been Completed, or Not Started.
   - The user who created the analysis.
   - A menu for submitting or scheduling the analysis.
7. You can search the list by analysis name in the search field. You can click Show Filters to access other filters, including analysis Status, and Autoupdate setting.
8. Click Create to create a new analysis.
9. On the Scope tab of the new analysis, enter a name and optional description.
10. Leave Autoupdate set to No yet. You cannot change it back to No once you have run the analysis.
11. Select Select and Add from the Actions menu, to select the items to be in the scope of the analysis. If you select an item class, then the search fields are adjusted to include fields from associated attribute groups.
   - To specify a scope of manually chosen items, select the items in the search results list and click Add Items. The items are added to the table of criteria in the scope of the analysis.
   - To specify a dynamic scope, add a saved search to the criteria table. In the Search and Select dialog box, select any existing saved search, or create a new saved search, and set it as default saved search. Then click Add Saved Search. When you later run the analysis, the saved search is executed. All the items that meet the search criteria at that point in time are considered as the scope of the impact analysis, and the analysis is run on this set of items.
12. When you have completed the scope criteria for the analysis, click Save.
13. Click Submit to run the Analyze Item Rule Set Impact process immediately. You can alternatively click Schedule to open the Schedule dialog and specify a schedule for when the analysis process runs.

When the analysis process runs, the draft rules included in the master rule set are force-applied to the items in the scope of the analysis, along with all the production (non-draft) rules included in the master rule set.

Reviewing Impact Analysis Results

When an analysis process completes successfully, you can open the analysis and review the results, on the Results subtab, which contains the affected items and any changes that would be made by the rules.

On the landing page of the Product Information Management work area, you can review impacts at a high level, using the Rules Impact Analysis infolet:

- By default, the infolet displays graphics linked to items impacted by runs of analysis processes during the Time Interval displayed at the top of the landing page. In the infolet, you can click the menu link for Items Impacted to select Items Not Impacted.
- A bar chart indicates the number of items impacted, or not impacted, for each impact analysis. The names of the analyses are keyed to the bars. Hovering your mouse over a bar displays the name of the analysis and the number of items.
- Clicking on a bar opens the Results tab of the Edit Impact Analysis page for the linked analysis.

On the Edit Impact Analysis page, you can review the details of the selected analysis, and filter the list of items in its scope in different ways, to better understand the impact.

- The Results tab displays the results of the analysis for all of the items specified in the scope of the analysis on the Scope tab. You can filter the list of items.
- Selecting one of the following infotiles enables you to work with the different categories of impacts:
  - **Impact**: Displays the number of items that are impacted or not impacted by the draft rules in the analysis.
    - Infotile filters: Impacted Items, Nonimpacted Items
    - Item results filters: Assignment Type (for associated assignment rules), Severity (for validation rules), Rule Name (for an analysis that contains multiple rules)
    - Item results actions: Export to Excel
  - **Rule Type**: Displays the number of items associated with either assignment rules or validation rules.
    - Infotile filters: Assignment, Validation
Item results filters: Assignment Type, Rule Name
Item results actions: Export to Excel

- **Actionable**: Displays the number of items that require action. Items can be either ready to add to a batch for import, or need corrections before being imported. If items need corrections, you can edit the items in a spreadsheet or use other means to make the corrections, then reload them for import.

Infotile filters: Ready, Need Corrections
Item results filters: Assignment Type, Severity, Rule Name
Item results actions: Export to Excel, Add Items to Batch, Edit Items in Spreadsheet

To use the Edit Items in Spreadsheet action, you must have installed ADFdi on your computer. From the Navigator link, select **Download Desktop Integration** in the Tools section.

- You can filter the list by the **Assignment Type** (for associated assignment rules), **Severity** (for validation rules), and **Rule Name** (for an analysis that contains multiple rules).
- For each item in the item results list, you can display a variety of fields, including:
  - Attribute values before and after being changed by assignment rules
  - New organizations and catalogs or categories assigned by rules
  - User messages caused by validation rule sets
  - The name of the rule that impacted the item

**Scenarios for Post-Analysis Update**

After you review the results of an impact analysis, you can rectify errors caused by rules, if any, by editing the items in a spreadsheet or using other means to make the corrections, then reload the items for import.

Consider an example scenario in which you have modified an assignment type item rule to reflect a pending change in a local sales tax. The rule will be applied to any new items that are created, but many items already exist that were created under the previous tax. You need to analyze the impact of applying the new tax assignment to those items.

To resolve the effects of the rule analysis:

1. Click the **Actionable** infotile. You will be determining the action to take before the impacted items are ready for import back into production.
2. Take the rule sets that produced the impact out of draft status. You can click the name of a rule set in the item results list to access the Edit Rule Set page. Deselect the **Draft** check box to take the rule set out of draft status.
3. The actions you can take are determined by the category of actionable items that you choose: **Ready** or **Need Corrections**. Use one of the following scenarios.

   - **Ready**: These items are likely to be imported with no errors produced by the impact of the rules in the analysis set. Suggested actions include:
     - You can add these items to a new or existing batch so that they can be imported into production.
     - Action: Select **Add Items to Batch** from the Actions menu. If you select **New Batch**, make selections in the **Process Details** dialog. The resulting rules update impact type of batch is predefined as an import to the Product Information Management Data Hub spoke system, which means that it updates only items that already exist. You can also add items to an existing rules update impact batch that was previously created by this option. You can examine the results of the batch import on the Manage
Item Batches page of the Product Information Management work area. You can examine the record of the import process in the Scheduled Processes work area, using the process ID displayed when you submitted the batch.

- Optionally, you can add the items to a spreadsheet, to further prepare them for import.
  
  Action: Select **Edit Items in Spreadsheet** from the Actions menu.

  **Need Corrections**: These items are likely to have errors when imported that are produced by the impact of the rules in the analysis set. For example, import errors would be caused by the triggering of rules with Reject severity.

Rules that cause errors typically require an you to make a decision on how to correct the data. For example, if there is a rule which says Minimum Temp should be greater than Maximum Temp, the item data could be corrected by correcting either the Minimum Temp or the Maximum Temp attribute values.

Suggested actions include:

- You can add the items impacted by a rule to a spreadsheet, to edit them to be suitable for import when impacted by a rule.
  
  Action: Select **Edit Items in Spreadsheet** from the Actions menu. In the **Edit Items in Spreadsheet: Select Rule** dialog box, select the rule set and rule that impacted the items that need correction, then click **Download**. The spreadsheet opens on your local computer, containing the selected items. After you edit the item data, you can import it.

- If you determine that corrections are needed because the impacting rule is not acting as expected, then you can edit the rule instead of the item data. In a row for a selected item, click the **Edit Rule** control to edit the rule that produced the impact shown on that item. To see which other items are also impacted by the same rule, use the Rule Name filter on the results list, which filters the list to the items affected by the same rule.

  After you make the needed corrections to the item data or to the rules, you can select and add these items to a new or existing batch so that they can be imported into production.

  Action: Select **Add Items to Batch** from the Actions menu.

---

**Related Topics**

- Item Batch Options: Explained

---

**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 Customer Relationship Management 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.

```
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.

### Assignment and Validation Examples

#### Assigning Organizations Using Item Rules: Example

You can define assignment rules that automatically assign items to one or more organizations when a condition is satisfied. The condition can be based on attribute values, another organization assignment, or a catalog assignment.

**Scenario**

The following table summarizes an example of an item rule that:

- Is defined in an assignment rule set.
- Tests whether an item has a product type of In-house. If the result of the test is true, then assign the item to the organizations M2 and M3.
Tests whether an item has a product type of *Bought-outside*. If the result of the test is true, then assign the item to the organization T1.

Note that this rule has more than one THEN expression. You add THEN expressions by using the Add Row action. The expressions are evaluated in the sequence of the rows, and the execution is halted when the first THEN expression evaluates to True. You can add additional rows of Then Expressions by clicking Add Row.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Organization assignment</td>
</tr>
<tr>
<td>Description</td>
<td>Auto assignment of item to organization based on given attributes.</td>
</tr>
<tr>
<td>Return type</td>
<td>Organization</td>
</tr>
<tr>
<td>Primary If Expression</td>
<td>true</td>
</tr>
<tr>
<td>Secondary If Expression</td>
<td></td>
</tr>
<tr>
<td>Return Value</td>
<td>Organization codes must be entered in quotation marks, either single, or double.</td>
</tr>
<tr>
<td>Secondary If Expression</td>
<td></td>
</tr>
<tr>
<td>Return Value</td>
<td>'T1'</td>
</tr>
<tr>
<td>User Message</td>
<td>The item was assigned to an organization, based on the product type.</td>
</tr>
</tbody>
</table>

Validating Organizations Using Item Rules: Example

You can define validation rules that automatically validate the assignment of items to one or more organizations when a condition is satisfied. The condition can be based on attribute values, another organization assignment, or a catalog assignment.

Scenario

The following table summarizes an example of an item rule that:

- Is defined in a validation rule set that is associated with an item class or with an attribute group.
- Following the assumption that items of the item class Extra Data Servers must not be assigned to the Seattle Distribution Center (code S2), validates that an item is not assigned to S2 if its item class is Extra Data Servers. This validation is performed by the following steps:
  - Test whether the item class of an item is *Extra Data Servers*.
  - If the result of the test is true, then test whether the item is not assigned to the organization S2, by using the function `assignedtoOrg(orgCode)`, which returns `true` if the item is assigned to the organization specified by `orgCode`.
Implementing Product Management

Chapter 6

Defining Product Rules

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Organization validation</td>
</tr>
<tr>
<td>Description</td>
<td>Validate assignment of Extra Data Servers</td>
</tr>
<tr>
<td>Severity</td>
<td>Reject</td>
</tr>
<tr>
<td>IF Expression</td>
<td><code>[Item].[Item Basic].[Item Class] == 'Extra Data Servers'</code></td>
</tr>
<tr>
<td>Validation Condition</td>
<td><code>!assignedToOrg('S2')</code></td>
</tr>
<tr>
<td>User Message</td>
<td>Extra Data servers cannot be assigned to the Seattle Distribution Center.</td>
</tr>
</tbody>
</table>

Defining Change Policies with Validation Rules: Examples

You can define validation rules that automatically enforce change policies by testing for prohibited changes and setting the severity of the validation.

Scenario

The following tables summarize an example of an item rule set and two item rules that enforce a pair of change policies on item structures.

<table>
<thead>
<tr>
<th>Rule Set Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Entity</td>
<td>Item</td>
</tr>
<tr>
<td>Rule Set Name</td>
<td>Structure Validation Policy</td>
</tr>
<tr>
<td>Rule Set Type</td>
<td>Validation</td>
</tr>
<tr>
<td>Association Type</td>
<td>Item Class</td>
</tr>
<tr>
<td>Association Name</td>
<td>Root Item Class</td>
</tr>
</tbody>
</table>

The first rule:

- Tests whether the structure name for the item is Manufacturing.
- If the test for the structure name evaluates to true, performs the validation specified by the Validation Condition, which is the expression false.
- If the Validation Condition evaluates to false, then the validation fails, and the action for the Severity is performed, which is Needs Approval.
Consequently, if a change is being attempted to the Manufacturing structure of the item, then the rule forces you to create a change order and get it approved.

<table>
<thead>
<tr>
<th>Rule Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Structure Policy Rule 1</td>
</tr>
<tr>
<td>Description</td>
<td>Ensures that changes to the Manufacturing structure of the item require approval of a change order.</td>
</tr>
</tbody>
</table>

**Valid Component Rules** check box | Check box must be deselected.

<table>
<thead>
<tr>
<th>Severity</th>
<th>Needs Approval</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF Expression</td>
<td><code>[Structure].[Structure Attributes].[Structure Name] == &quot;Manufacturing&quot;</code></td>
</tr>
<tr>
<td>Validation Condition</td>
<td><code>false</code></td>
</tr>
<tr>
<td>User Message</td>
<td>Changes to this structure require the creation and approval of a change order.</td>
</tr>
</tbody>
</table>

The second rule:

- Tests whether the structure name for the item is *Engineering* and the Life Cycle Phase is *Design*.
- If the test for the structure name evaluates to true, performs the validation specified by the Validation Condition, which is the expression `false`.
- If the Validation Condition evaluates to false, then the validation fails, and the action for the Severity is performed, which is **Reject**
- Consequently, if a change is being attempted to the Engineering structure of the item while the Life Cycle Phase is *Design*, then the rule rejects the change and prevents it from occurring.

<table>
<thead>
<tr>
<th>Rule Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Structure Policy Rule 2</td>
</tr>
<tr>
<td>Description</td>
<td>Ensures that changes to the Engineering structure cannot be made if the life cycle phase is Design.</td>
</tr>
</tbody>
</table>

**Valid Component Rules** check box | Check box must be deselected.

<table>
<thead>
<tr>
<th>Severity</th>
<th>Reject</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF Expression</td>
<td><code>[Item].[Main].[Life Cycle Phase]==&quot;Design&quot;</code></td>
</tr>
<tr>
<td>Validation Condition</td>
<td><code>false</code></td>
</tr>
<tr>
<td>User Message</td>
<td>Changes to this structure cannot be made during the Design life cycle phase.</td>
</tr>
</tbody>
</table>
Changing Item Supplier Site Organization Associations Using Validation Rules: Examples

You can define validation rules that automatically govern the creation and updating of item supplier site organization associations and the updating of association attributes, both Primary attributes and extensible flexfields.

Scenario

The following tables summarize an example of an item rule set and rules that govern an item supplier site organization association.

<table>
<thead>
<tr>
<th>Rule Set Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Entity</td>
<td>Item</td>
</tr>
<tr>
<td>Rule Set Name</td>
<td>Supplier Association Rule Set</td>
</tr>
<tr>
<td>Rule Set Type</td>
<td>Validation</td>
</tr>
<tr>
<td>Association Type</td>
<td>Item Class</td>
</tr>
<tr>
<td>Association Name</td>
<td>Electronics</td>
</tr>
</tbody>
</table>

The rule:

- Is applied when an item belongs to the item class Electronics.
- Because the IF expression is true, performs the validation specified by the Validation Condition.
- The Validation Condition tests whether name of the item supplier site is Acme.
- If the Validation Condition evaluates to true, then the validation succeeds, and the action for the Severity is performed, which is Needs Approval
- Consequently, if the user tries to add an item supplier site association for the supplier Acme to an item belonging to the item class Electronic, then the rule forces you to create a change order and get it approved.

<table>
<thead>
<tr>
<th>Rule Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Supplier Association Rule</td>
</tr>
<tr>
<td>Description</td>
<td>Ensures that a change order is required to add an item supplier site association for the supplier Acme.</td>
</tr>
<tr>
<td>Valid Component Rules check box</td>
<td>Check box must be deselected.</td>
</tr>
<tr>
<td>Severity</td>
<td>Needs Approval</td>
</tr>
<tr>
<td>IF Expression</td>
<td>true</td>
</tr>
</tbody>
</table>
Defining Product Rules

Validating Component Types Using Rules: Examples

You can define validation rules that automatically check whether the components associated with the structure are valid. You can specify the list of valid user item types for components in the rule.

**Scenario**

The following table summarizes an example of an item rule set and item rule that:

- Tests whether the structure associated with the item is of type Primary.
- Validates that only components of item types \texttt{type1} and \texttt{type2} can be added to a structure of type Primary.

<table>
<thead>
<tr>
<th>Rule Set Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Entity</td>
<td>Item</td>
</tr>
<tr>
<td>Rule Set Name</td>
<td>Validate Component Types</td>
</tr>
<tr>
<td>Rule Set Type</td>
<td>Validation</td>
</tr>
<tr>
<td>Association Type</td>
<td>Item Class</td>
</tr>
<tr>
<td>Association Name</td>
<td>Root Item Class</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Validate Component Types for Primary Structure</td>
</tr>
<tr>
<td>Description</td>
<td>Ensures that only components of item type TYPE1 or TYPE2 can be added to a Primary structure.</td>
</tr>
</tbody>
</table>

**Valid Component Rules** check box

Check box must be selected.

Unlike other validation rules, rules that validate components do not have a Severity, because the item is being validated for only one aspect, and a degree of severity would be redundant.

<table>
<thead>
<tr>
<th>IF Expression</th>
<th>Validation Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>component_type(&quot;type1&quot;,&quot;type2&quot;)</td>
<td></td>
</tr>
</tbody>
</table>
Validating Item Class Changes with Rules: Examples

You can define validation rules that automatically check whether the item class of an item has been changed.

Scenario

End users are able to change the item class for an item. You can use a validation rule to restrict whether such a change is to, or from, a specific item class. Use the Severity value of the rule to warn about the change, reject the change, or require that the change needs an approved change order.

The following tables summarize an example of an item rule set and item rule that:

- Tests whether the associated item is being changed to the item class TCCHLDVR, and also changed to the item class TCParent.
- If the test for item class changes evaluates to true, perform the validation specified by the Validation Condition, which is the expression false.
- If the Validation Condition evaluates to false, then the validation fails, and the action for the Severity is performed, which is Reject.
- Consequently, if the item class is being changed to TCCHLDVR and also changed from TCParent, then the change is rejected by the rule.

<table>
<thead>
<tr>
<th>Rule Set Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Entity</td>
<td>Item</td>
</tr>
<tr>
<td>Rule Set Name</td>
<td>Validate Item Class</td>
</tr>
<tr>
<td>Rule Set Type</td>
<td>Validation</td>
</tr>
<tr>
<td>Association Type</td>
<td>Item Class</td>
</tr>
<tr>
<td>Association Name</td>
<td>TCParent</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Validate Source and Target IC</td>
</tr>
<tr>
<td>Description</td>
<td>Ensures that items cannot change item class from TCParent to TCCHLDVR.</td>
</tr>
<tr>
<td>Severity</td>
<td>Reject</td>
</tr>
</tbody>
</table>
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 spoke systems, specific attributes of the item might have different values, depending on the spoke system. To control which attribute value is imported into production, you can define blending rules, which are applied during import, and which determine which spoke system’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 spoke system 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. 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 spoke system 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 spoke system provides updates to product data that was imported earlier.
- When a new spoke system 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 spoke systems mentioned in the blending rule has provided data into the supplier stage. If the higher priority spoke system has not provided any data, then whatever is provided by the lower priority or other spoke systems (spoke systems 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. Blended data 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.
- When you use an import map and upload the data into an internal spoke system batch, the data is inserted in the staging area as well as in the Product Hub batch interface. Pre-processing or import is initiated on the batch according to the spoke system setup.
- You can enable synchronization of null attribute values from the staging area, to the batch interface area, and then to the production area. When creating a blending rule, if you deselect the Ignore Null check box for a source system, then null values being imported replace the corresponding values that exist for the attribute in production.
  - If an attribute group is a multi-row extensible flexfield, then the missing rows in the uploaded product data are removed from the flexfield in production.
  - If an attribute group is used in a single-row extensible flexfield, then the attributes with null values in the uploaded product data are also replaced with nulls in the flexfield in production.
  - During the import process, null values from staging are represented by placeholders in the batch interface area, such as #NULL for character values.

For any synchronization of attributes between a spoke system and an item to happen, with null values or otherwise, the item must include a a spoke system item relationship to the spoke.

Restrictions and validations on blending rules include:

- Synchronization of attribute values is not performed for seeded operational item attributes.
- 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.
• Spoke systems categorized as internal systems are available as source systems for use in blending rules. The predefined Product Information Management Data Hub spoke system is not available as a source system for use in blending rules.
• For spoke systems categorized as internal systems, only data uploaded using an import map will be available for blending.
• 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.
   - In the Association Name list, search for and select the name of the entity that contains the attribute whose value you want to control with blending rules.
     - 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:
> 1. **Return Type**
> 2. **Target Business Entity**
> 3. **Target Attribute Group**
> 4. **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. You can deselect the **Ignore Null** check box to enable synchronization of attributes. By default, the check box is selected, to ignore nulls in the staging data. Accordingly, attributes are not synchronized unless you choose to do so.

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 **BLENDING_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 **Add** from the Actions menu.
4. 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.

5. Select the rules sets you want to add, then click **OK** to add them to the master blending rule set.

6. 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.

7. 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**
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
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 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 (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
  o Spoke Item:
    • Vendor1 TV Item
      o Attribute: Frame Description
        • Value: Plastic body of PVC make with metallic back hinge (updated)
      o Attribute: Resolution
        • Value: 1930x1024
      o Attribute: Height (new)
        • Value: 2 Feet (new)
  • Vendor2
    o Spoke Item:
      • Vendor2 Television
      o Attribute: Frame Description
        • Value: PVC body with hinge (updated)
      o Attribute: Resolution
        • Value: 2930x1034
      o Attribute: Weight (new)
        • Value: 5 kg (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:
  o Production item in Product Hub:
    • TV_1
      o Spoke Item cross-references:
        • Vendor1 TV Item
        • Vendor2 Television
      o 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 Defining 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

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

The 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. You can add a role (that includes list of users) as an assignee for the NIR. When the role expanded into a comma separated list, the number of characters in the list should not exceed 2000.

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

- Offering: Product Management
- Functional Area: New Item Requests
- Task: Manage New Item Request Type Details

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.

You can also add a role (that includes a list of users) as an approver. When the role expanded into a comma separated list, the number of characters in the list should not exceed 2000.

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, see the Oracle Cloud Developing SOA Applications with Oracle SOA Suite guide.

*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
- Change Order Approval Process: Explained

**Troubleshoot Access to New Item Requests: Explained**

If the user is unable to view or manage NIRs, ensure that the relevant privileges are added to the role.

The table lists the privileges required to view and manage NIRs, and the corresponding privilege code.

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Privilege Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approve New Item Request</td>
<td>EGO_APPROVE_NEW_ITEM_REQUEST_PRIV</td>
</tr>
<tr>
<td>Manage New Item Request</td>
<td>EGO_MANAGE_NEW_ITEM_REQUEST_PRIV</td>
</tr>
<tr>
<td>View New Item Request</td>
<td>EGO_VIEW_NEW_ITEM_REQUEST_PRIV</td>
</tr>
<tr>
<td>Monitor New Item Request Summary</td>
<td>EGO_MONITOR_NEWITEMREQUEST_SUMMARY_PRIV</td>
</tr>
<tr>
<td>Item Inquiry</td>
<td>ORA_EGP_ITEM_INQUIRY_DUTY</td>
</tr>
</tbody>
</table>

To edit the role, use Security Console.

**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.</td>
</tr>
<tr>
<td></td>
<td>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. Run the **Manage Item Rule Sets** task (in the Setup and Maintenance work area and the Product Management offering).
2. Create a rule. In the Create Rule Set dialog, you must make the following selections.
   a. For Type, select **Validations**.
   b. For Association Type, select **New Item Request**.
   c. 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 `== "Cost"`, to form the following complete comparison expression:
   
   \[[\text{ChangeHeader}]> [\text{ChangeHeaderMain}]> [\text{ReasonCode}] == "Cost"\]

6. In the **Validation Condition** field, use the same procedure to insert the following expression:
   
   \[[\text{ChangeHeader}]> [\text{ChangeHeaderMain}]> [\text{PriorityCode}]=="High"\]

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. In the Setup and Maintenance work area, use the following:
   - Offering: Product Management
   - Functional Area: New Item Requests
   - Task: Manage New Item Request Type Details

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
- Item Rule Syntax: Explained
8 Defining 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:**
  
  `INVENTORY_ITEM_STATUS_CODE == 'Active'`

- Item Validation Rules are evaluated prior to publishing.
9 Defining 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.
Chapter 10

Defining 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>


11 Defining Imports

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.

Item Imports: Explained

You can import items and item-related information using interface tables. This import data is loaded into the production tables using the Import Item task.

Import Item

The Item Import task creates an Enterprise Storage Server (ESS) process that takes the data that is loaded in the interface tables and uses the import process to move the data to the production tables. The import processes will perform all of the validations necessary to ensure the data imported is correct prior to moving the data into the production tables.

1. Access the Enterprise Storage Server and provide a process name (job definition) such as Item Import Process.
2. Select Setup and Maintenance from the Navigator.

Access the All Tasks tab on the Overview page, and search for the Import Item task with the name of your ESS process definition.
3. Click the Go to Task icon in the search results for that Import Item task.

The parameters for the item import process are:

- **Batch ID**: Associate the interface table to an item batch definition.
- **Organization**: Select an organization to be used for the import.
Process Only: Determines how the data is processed. The choices are:
- Create
- Sync
- Update

Process All Organizations: Select Yes if the import contains items that will be imported to multiple organizations.

Delete Processed Rows: Select Yes to delete rows that are imported without errors.

4. Click Submit and the Request Number will be displayed.

Deleting Values by Import

You can use the import process to delete values of attributes, including individual values of multi-row attributes.

1. Download the template spreadsheet, ItemImportTemplate.xlsx.
2. Follow the instructions contained in the spreadsheet to modify attribute values. Select the Transaction Type of DELETE for the values that you want to delete.
3. Follow the instructions contained in the spreadsheet to generate a CSV and Zip file.
4. Select File Import and Export from the Navigator.
5. Create an upload for the template spreadsheet, using the account scm/item/import.
6. Create a scheduled process using the job Load Interface File for Import. In the Process Details, select Item Import for Import Process and the uploaded Zip file for Data File.

Monitoring Import Items

Select Setup and Maintenance from the Navigator, then access Monitor Item Imports to search for specific Enterprise Storage Server processes and monitor their status in the search results table.

Related Topics
- Manage Imports: Overview
- Item Batches: Overview

Item Data Conversion: Points to Consider

To support data conversion and initial load to Product Hub, the profile option: Item Data Conversion Mode Enabled must be set to enabled.

When you perform data conversion from your legacy applications into Product Hub, you must set the profile option value to Yes. The default value is No. Setting the profile value to Yes will have the following effects:

- The Import process ignores all security checks. Data Privilege and Functional privileges are not considered.
- All items will be created as approved items irrespective of whether the item class is enabled for new item request or you select to Add All Items to a new item request in batch.
- All updates will be directly posted to production by not creating change orders, even if you select the Add All Items to change order option in the batch.
• All validation rules of type Needs Approval will be ignored. This means that items are updated in the production tables.

If the profile option values are incorrect, it can adversely affect performance. You should evaluate carefully before setting these options.

### Generate the CSV File: Explained

After downloading the template, enter data into the item import 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 item import 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 item import template 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 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
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 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.

Related Topics
- Import Items: Explained

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.
Importing an Item Structure: Worked Example

Importing structures of items is similar to importing regular items; however, there are enough exceptions and wrinkles to take the Item Structure import process from the beginning.

This is an overview to importing item structures:

1. Download the Item Structure Import template file from the File-Based Data Import (FBDI) for Oracle Supply Chain Management Cloud.
2. Enter data in tabs within the Item Structure Import template file.
3. Generate a CSV (ZIP) file.
4. Upload to your Oracle SCM Cloud application.
5. Move the data into Item Management interface tables.
6. Import items, then import item structures to Item Management product tables.

Preliminary

When you intend to collect items in a structure and you want to import that structure, the regular items must first exist in the application. If some or all items going into the structure do not exist already in the application, these items must be created or imported before you design and import the structure.

You will be working with an Excel spreadsheet named Item Structure Import template.xlsm. You can download the template from http://docs.oracle.com by navigating to: Cloud > Applications > SCM Core > Use > Import file-based data > Item Structure.

The tabs (or pages or sheets) in the spreadsheet are as follows:

- Instructions and CSV Generation - use for reference
- EGP_STRUCTURES_INTERFACE
- EGP_COMPONENTS_INTERFACE
- EGP_SUB_COMPS_INTERFACE
- EGP_REF_DESGS_INTERFACE

On the four functional sheets, the key column heads are:

- Transaction Type
- Batch ID
- Batch Number
- Structure Name
- Common Structure Name
- Organization Code
- Common Structure Organization Code
- Item Name
Setting Up the Spreadsheet to Design the New Structure

You must plan and decide exactly which items shall become structures, as well as quantities of each item that shall be required on any specific structure.

1. Open the Item Structure Import spreadsheet template.
2. Use the EGP_STRUCTURES_INTERFACE tab to designate which items will be included in the structure.
3. On the EGP_COMPONENTS_INTERFACE tab, designate the Relationships that each item has with any other item in the structure, parent-to-child or child-to-parent. List the child items in column G, Component Item Name. List the respective parent items in column F, Structure Item Name.
4. Use column I, Sequence, to define the order in which the items will be listed in the structure.

Uploading the Spreadsheet Data to the Interface Tables

You have set up the spreadsheet for your import process, now you will go to your Oracle Cloud application.

1. Return to the first tab in the spreadsheet template: Instructions and CSV Generation. Click the Generate CSV File button. You are prompted to select a location and file name for this ZIP file. The prompted filename is EgpStructuresImport.Zip. After choosing a location, click Save. Next, you are requested to save each CSV file inside that ZIP file.
2. In Oracle Cloud, use Navigator to open File Import and Export. When the page opens, click the Upload File + icon, click Browse button, and select EgpStructureImport.zip. Set SCMItemImport as the account. Click Save and Close.
3. In order for the import to proceed, the process must be scheduled. To schedule the import process, return to Navigator, click Scheduled Processes, and click the Schedule New Process button.
4. In the Schedule New Process: Name field, click Search in order to locate and select Load Interface File for Import, and click OK. In the Schedule New Processes dialog, click OK.
5. For the Process Details, select Item Import as the Import Process, and in the Data File dialog, select EgpStructuresImport.Zip.
6. Now click Submit. Either write down or copy the Process Number in the confirmation message, and click OK. (This is not a trivial precaution. Note that if you only store it in your copy memory, CTRL-c, you may easily overwrite that doing something else before you need the Process number.)
7. Refresh the search results, clicking the Refresh icon until the status has changed to Succeeded. The data is now loaded to the interface tables.

Completing the Import

Now the data must be imported from the interface tables into the item structure tables.

1. To schedule a new process, look through Search returns for Item Import. Select and click OK. Now insert the Batch ID, from the spreadsheet. (By inserting this Batch ID, the application fetches only those transactions. If you have other transactions on the same sheets with a different Batch ID, they are not fetched now. It fetches only those that you select or designate by the Batch ID.)
2. Click Submit, get the Process ID, then click OK, and it starts the process. After some time, click Refresh.
3. When the status has changed to Succeeded, all subprocesses have finished.
4. Look through the list of items to confirm that the structure was created, that is, that the item which is now a structure displays a Structure icon.
5. Schedule a process to purge the interface tables. Click **Schedule New Process**, search for and select **Purge Interface Tables**, and click **OK**. On the Schedule New Process dialog, click **OK** again.

6. On the **Process Details** dialog, confirm that Purge Process Intent is set to **File-based data import**, select **Item Import** as the Import Process, add the Load Request ID, this is the last Process ID that you copied or wrote down. Now click **Submit**.

**Importing Agile PLM Business Objects to Oracle Cloud: Details**

Other topics about Importing, such as "Importing an Item Structure: Worked Example", present procedures to migrate business objects outside Oracle Cloud to your Cloud applications, such as Product Master Data Management and Product Development. This topic gives supporting information about importing standard business objects - Items, Structures (BOM), and Change Orders - as well as other objects that may be present in your legacy Agile PLM suite of applications.

<table>
<thead>
<tr>
<th>Agile PLM Business Object or Component</th>
<th>Excel Templates</th>
<th>Template Tab</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ITEMS</strong></td>
<td>ItemImportTemplate.xlsm</td>
<td>EGP_SYSTEM_Items_INTERFACE</td>
<td>This carries all items and their standard attributes (therefore, does not include Page Two and Page Three attributes).</td>
</tr>
<tr>
<td>Revisions</td>
<td>ItemImportTemplate.xlsm</td>
<td>EGP_ITEMS_REVISIONS_INTERFACE</td>
<td>This carries the revisions (old and current) of all items. Each revision must include a valid Effectivity Date.</td>
</tr>
<tr>
<td>Structure (BOM)</td>
<td>ItemStructureImportTemplate.xlsm</td>
<td>EGP_STRUCTURES_INTERFACE</td>
<td>This carries the primary structures of all items.</td>
</tr>
<tr>
<td>Item Components</td>
<td>ItemStructureImportTemplate.xlsm</td>
<td>EGP_COMPONENTS_INTERFACE</td>
<td>This carries the components of all items. To import revision-specific structures, ensure that every component Effectivity Date aligns with the corresponding revision’s Effectivity Date.</td>
</tr>
<tr>
<td>Component Reference Designators</td>
<td>ItemStructureImportTemplate.xlsm</td>
<td>EGO_REF_DSGS_INTERFACE</td>
<td>This carries the reference designators of all components.</td>
</tr>
<tr>
<td>Approved Manufacturers List (AML) and Manufacturer Parts (MPN)</td>
<td>ItemImportTemplate.xlsm</td>
<td>EGP_TRADING_PARTNER_ITEMS_INTF</td>
<td>This carries all Manufacturer Parts. Manufacturers have been imported using a Customer Relationship Management (CRM) FBDI template. AML and MPN cannot be imported until the individual Manufacturers are imported.</td>
</tr>
<tr>
<td>Agile PLM Business Object or Component</td>
<td>Excel Templates</td>
<td>Template Tab</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-------------------------------------</td>
<td>--------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Page Two or Page Three</td>
<td>ItemImportTemplate. xlsm</td>
<td>EGO_ITEM_INTF_EFF_B</td>
<td>This carries the Page Two and Page Three attributes of all items. Use the Version Start Date to match the Effectivity Date of the item revision.</td>
</tr>
<tr>
<td>Structure EFfS</td>
<td>ItemStructureImportTemplate. xlsm</td>
<td>EGP_COMPONENTS_INTERFACE</td>
<td>This carries the EFf attributes of all structures.</td>
</tr>
<tr>
<td>Attachments</td>
<td>ItemImportTemplate. xlsm</td>
<td>EGP_ITEM_ATTACHMENTS_INTF</td>
<td>This carries the mapping data for Item ID, Revision Number, and Attachment ID.</td>
</tr>
<tr>
<td>CHANGE ORDERS</td>
<td>ChangeOrderImportTemplate. xlsm</td>
<td>EGO_CHANGES_INT</td>
<td>This carries all change orders, their standard attributes, and their global/ context DFFs.</td>
</tr>
<tr>
<td>Affected Items</td>
<td>ChangeOrderImportTemplate. xlsm</td>
<td>EGO_CHANGE_LINES_INT</td>
<td>This carries all affected items. Ensure that Import change lines and revisions are in chronological order. Ensure that the values in Scheduled Date and New Item Revision correspond to Effectivity Date and Revision (already imported through Item FBDI Import).</td>
</tr>
<tr>
<td>Change Order Attachments</td>
<td>ChangeOrderImportTemplate. xlsm</td>
<td>EGO_CHANGE_ATTACHMENTS_INTF</td>
<td>This carries the mapping data for Change Number and Attachment ID.</td>
</tr>
</tbody>
</table>

**Note:** Change orders that are still open in Agile PLM are not imported to Oracle Cloud application. Change orders must be completed - Released - in Agile PLM before they can be imported to Oracle Cloud.

**Related Topics**
- Importing an Item Structure: Procedure

### 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 Configuration</td>
<td>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.

Importing Items with FBDI: Explained

The set of Product Hub interface tables includes the main EGP_SYSTEM_ITEMS_INTERFACE table in which item data can be inserted and 13 other interface tables in which data of the item’s child entities such as revisions, categories, and extensible flexfields can be inserted.

You can use the control files provided to load data from a .csv file into interface tables using the Load Interface File for Import process.

<table>
<thead>
<tr>
<th>Control File</th>
<th>Interface Table</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>EgpSystemItemsInterface. ctl</td>
<td>EGP_SYSTEM_ITEMS_INTERFACE</td>
<td>Items</td>
</tr>
<tr>
<td>EgpItemRevisionsInterface. ctl</td>
<td>EGP_ITEM_REVISIONS_INTERFACE</td>
<td>Item Revisions</td>
</tr>
<tr>
<td>EgpItemCategoriesInterface. ctl</td>
<td>EGP_ITEM_CATEGORIES_INTERFACE</td>
<td>Item Categories</td>
</tr>
<tr>
<td>EgpItemAssociationsIntf. ctl</td>
<td>EGO_ITEM_ASSOCIATIONS_INTF</td>
<td>Item Associations</td>
</tr>
<tr>
<td>EgpItemRelationshipsIntf. ctl</td>
<td>EGP_ITEM_RELATIONSHIPS_INTF</td>
<td>Item Relationships</td>
</tr>
<tr>
<td>EgpItemIntfEffb. ctl</td>
<td>EGO_ITEM_INTF_EFF_B</td>
<td>Item level flexfields</td>
</tr>
<tr>
<td>EgpItemIntfEfftl. ctl</td>
<td>EGO_ITEM_INTF_EFF_TL</td>
<td>Item level flexfields</td>
</tr>
<tr>
<td>EgpItemRevisionIntfEffb. ctl</td>
<td>EGO_ITEM_REVISION_INTF_EFF_B</td>
<td>Item Revision flexfields</td>
</tr>
</tbody>
</table>
As the details of the mapping between the data of a source system and Product Hub might vary significantly based on the source system, the scope of these recommendations is limited to a best practices approach.

- Build an export file in the same format as the Product Hub interface described in the template. You will need to create staging tables and a spreadsheet that mirrors the tabs contained in this template. Then, a mapping needs to be decided as to which data of the source system goes into which Product Hub interface columns and tables. Based on the mapping, you should extract the relevant data from the source system. You can use SQL, ODI, or a similar tool to extract data into your staging tables.

- Your spreadsheet must have the same columns, and the columns must be in the same order, as in the template. The value in the first column is always the interface table name.

- Once you have extracted the data into your temporary spreadsheet, cut and paste the data into the relevant sheets provided in the template.

- For ID columns, you will need to develop a mapping between existing values and Oracle Fusion values. Use the implementation pages in the Setup and Maintenance work area to extract the identifier. For columns where you will need to use the Setup and Maintenance work area to get the Oracle Fusion values, the comments in the column header of the spreadsheet mentions the task name which you should navigate to in Setup and Maintenance work area.

- For key flexfields, you will also need to develop a way to map the values in the source system that correspond to the unique combinations in Product Hub. For key flexfield columns in the spreadsheet that require internal IDs, use the related Oracle Fusion implementation pages to export the internal IDs and the segment values in order to map to the source system values. The Items interface table has six columns that get their values from Key flexfields:
  - Process Supply Locator
  - Process Yield Locator
  - WIP Supply Locator
  - Sales Account
  - Expense Account
  - Asset Category
Importing Data

After you successfully load your data, you must submit the Item Import process to import the data into the application tables to create Items and its child entities.

To submit the Item Import process:

1. Navigate to Manage Item Batches task in the Product Information Management work area.
2. Make sure that appropriate Batch Options are set for the batch.
3. Navigate to the Scheduled Processes task.
4. Click on Schedule New Process and select Item Import Process.
5. Enter the value for Batch ID and submit the process.
6. Monitor the process in the Search Results section.
7. If the Item Import Process ends in error or warning, review the log file for details.

Correcting Errors

To correct import errors:

1. Click on the Manage Item Batches task in the Product Information Management work area.
2. Search for the batch for which the Item Import process was run.
3. Click on the batch name to navigate to batch details.
4. Review the errors for each item.
5. Select the item rows that are in error and click on Manage in Spreadsheet to export Item data to spreadsheet.
6. Once all the rows are corrected, resubmit the process by clicking Upload and then submit the Item Import process.
7. Repeat the submit and error correction steps in this section until all rows are imported successfully and the items along with their child entities are created.

Import Map: Overview

Suppliers send product and catalog data to retailers and manufacturers in their native format, usually as spreadsheets or xml file format. They typically use standard industry formats, such as BMECat or eClass, or others. Retailers need to onboard or upload the data provided by suppliers into Oracle Fusion Product Hub.

To import data follow these steps:

- Select or create a batch.
- Add items to the batch.
- Select Upload from File or Upload from Spreadsheet.
- In this example, Upload from File is selected because the user has external system data in CSV format.
- In the Upload from File window the user reviews the batch number, batch name, and source system information.
- Select or create an Import Map.
- Review map and submit, which will load the data file into the interface tables, dismisses the dialog, and gives control back to the Manage Item Batches page.

FAQs for Defining Imports
What kind of item relationships can I create with FBDI?

Item relationships such as Item Cross Reference, Related Item, Trading Partner Items (Competitor, Customer, Manufacturer Part Number), and GTIN Cross Reference can be created. The value entered in the Item Relationship Type column determines the type of relationship that can be created.

How do I create Trading Partner Items with FBDI?

The EGP_TRADING_PARTNER_ITEMS_INTF worksheet must be populated with the following details:

<table>
<thead>
<tr>
<th>Column</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trading partner Item Number</td>
<td>Trading partner item number</td>
</tr>
<tr>
<td>Trading partner Item Description</td>
<td>Trading partner item description</td>
</tr>
<tr>
<td>Trading Partner Type</td>
<td>Type of trading partner item (MANUFACTURER, COMPETITOR, or CUSTOMER)</td>
</tr>
<tr>
<td>Trading Partner Number</td>
<td>Trading partner number</td>
</tr>
<tr>
<td>Item Relationship Type</td>
<td>CUSTOMER_ITEM_XREF</td>
</tr>
<tr>
<td>Item Number</td>
<td>Product Hub item number</td>
</tr>
<tr>
<td>Trading Partner Item Number</td>
<td>Manufacturer, competitor, or customer item number</td>
</tr>
<tr>
<td>Description</td>
<td>Manufacturer, competitor, or customer item number</td>
</tr>
</tbody>
</table>

*Note:* All other mandatory columns must populated.

How do I import a GTIN cross reference with FBDI?

The EGP_ITEM_RELATIONSHIPS_INTF worksheet must be populated with the following details:

<table>
<thead>
<tr>
<th>Column</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item Relationship Type</td>
<td>GTIN</td>
</tr>
<tr>
<td>Sub Type</td>
<td>Party Type (Customer, Manufacturer, or Supplier)</td>
</tr>
<tr>
<td>Cross Reference</td>
<td>Enter the GTIN number</td>
</tr>
<tr>
<td>EPC GTIN Serial</td>
<td>Enter the electronic product code GTIN Serial Number</td>
</tr>
</tbody>
</table>
How do I import a Customer Item relationship item with FBDI?

The EGP_ITEM_RELATIONSHPES_INTF worksheet must be populated with the following details:

<table>
<thead>
<tr>
<th>Column</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item Relationship Type</td>
<td>CUSTOMER_ITEM_XREF</td>
</tr>
<tr>
<td>Item Number</td>
<td>Enter the Product Hub item number.</td>
</tr>
<tr>
<td>Trading Partner Item Number</td>
<td>Customer item number</td>
</tr>
<tr>
<td>Description</td>
<td>Customer item description</td>
</tr>
</tbody>
</table>

*Note:* All other mandatory columns must populated.

How do I import a MPN relationship item with FBDI?

The EGP_ITEM_RELATIONSHPES_INTF worksheet must be populated with the following details:

<table>
<thead>
<tr>
<th>Column</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item Relationship Type</td>
<td>MFG_PART_NUM</td>
</tr>
<tr>
<td>Item Number</td>
<td>Enter the Product Hub item number.</td>
</tr>
<tr>
<td>Trading Partner Item Number</td>
<td>MPN number</td>
</tr>
<tr>
<td>Description</td>
<td>MPN description</td>
</tr>
</tbody>
</table>

*Note:* All other mandatory columns must populated.
How do I import a competitor item relationship with FBDI?

The EGP_ITEM_RELATIONSHIPS_INTF worksheet must be populated with the following details:

<table>
<thead>
<tr>
<th>Column</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item Relationship Type</td>
<td>COMPETITOR_ITEM_XREF</td>
</tr>
<tr>
<td>Item Number</td>
<td>Enter the Product Hub item number.</td>
</tr>
<tr>
<td>Trading Partner Item Number</td>
<td>Competitor item number</td>
</tr>
<tr>
<td>Description</td>
<td>Competitor item description</td>
</tr>
</tbody>
</table>

*Note:* All other mandatory columns must populated.

How do I import a related item with FBDI?

The EGP_ITEM_RELATIONSHIPS_INTF worksheet must be populated with the following details:

<table>
<thead>
<tr>
<th>Column</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item Relationship Type</td>
<td>ITEM_XREF</td>
</tr>
<tr>
<td>Description</td>
<td>Enter a relationship description.</td>
</tr>
<tr>
<td>Related Item Number</td>
<td>Item number</td>
</tr>
<tr>
<td>Sub Type</td>
<td>Related item type. The values for this can be searched from the lookup EGP_RELATIONSHIP_TYPE.</td>
</tr>
<tr>
<td>Start Date</td>
<td>Enter the start date.</td>
</tr>
<tr>
<td>End Date</td>
<td>Enter the end date.</td>
</tr>
<tr>
<td>Reciprocal</td>
<td>Enter Y or N.</td>
</tr>
<tr>
<td>Planning Enabled</td>
<td>Enter Y or N.</td>
</tr>
</tbody>
</table>

*Note:* All other mandatory columns must populated.
12 Defining 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 it’s 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.
13 Defining 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.
Related Topics

- Defining Product Innovation: Overview

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 configuration 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>Configuration Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Requirements and Ideation Management</td>
<td>Ideas</td>
<td>Status</td>
<td>Pending, Accepted, Rejected, Implemented, In Progress, Review</td>
<td>User</td>
</tr>
<tr>
<td>Product Requirements and Ideation Management</td>
<td>Requirements</td>
<td>Associated Product</td>
<td>Common Services, Product Concept Design, Product Lifecycle Portfolio Management, Product Requirements and Ideation Management</td>
<td>User</td>
</tr>
<tr>
<td>Product Requirements and Ideation Management</td>
<td>Requirements</td>
<td>Requirement Fulfillment</td>
<td>Yes, No</td>
<td>User</td>
</tr>
<tr>
<td>Product Requirements and Ideation Management</td>
<td>Requirements</td>
<td>Requirement Priority</td>
<td>Must Have, Nice to Have, Should Have</td>
<td>User</td>
</tr>
</tbody>
</table>
### Table: Lookups in Product Lifecycle Management

<table>
<thead>
<tr>
<th>Application</th>
<th>Module</th>
<th>Lookup Type</th>
<th>Lookup Code Meaning</th>
<th>Configuration 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.
- Each timeline in the Schedule, Resource, and Launch charts is determined by the number of planning period units that are 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 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>On Edit Portfolio page &gt; Actions &gt; Define Metrics leads to Define Metrics page, with Product Metrics tab.</td>
</tr>
<tr>
<td>Portfolios</td>
<td>On Edit Portfolio page &gt; Actions &gt; Define Metrics leads to Define Metrics page, with Product Portfolios tab.</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)

**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
- Configuring Oracle Innovation Management Cloud: Checklists

**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.</td>
</tr>
<tr>
<td></td>
<td>predefined business rules</td>
</tr>
<tr>
<td></td>
<td>Logical expression</td>
</tr>
<tr>
<td></td>
<td>Validation condition</td>
</tr>
<tr>
<td></td>
<td>User message</td>
</tr>
<tr>
<td>Assignments</td>
<td>Define the value of an attribute, based on the specified condition. Rules are executed in the order of their sequence in the rule set.</td>
</tr>
<tr>
<td></td>
<td>Target business entities are:</td>
</tr>
<tr>
<td></td>
<td>- Portfolio General Information</td>
</tr>
<tr>
<td></td>
<td>- Product</td>
</tr>
<tr>
<td></td>
<td>- Scenario</td>
</tr>
<tr>
<td>Composite</td>
<td>Aggregate rules sets that operate on different attribute groups. Composite rule sets contain both validation and assignment rule sets.</td>
</tr>
</tbody>
</table>
Related Topics

- Rules and Rule Sets: Explained

Define Product Development

Product Development Components: How They're Configured

This topic describes the hierarchy of tasks required to set up 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
- Change Order Approval Process: Explained
- Revisions: Explained
- Setting up Product Development: Roadmap

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 configuration 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>Configuration 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>Configuration 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 in change reasons, run the Manage Change Reasons task in the Product Management offering (available in the Setup and Maintenance work area).

<table>
<thead>
<tr>
<th>Common References</th>
<th>Common</th>
<th>Class Family Name</th>
<th>Design, Concept, Concept Component, Portfolio, GSCC Placeholder, Idea, Requirement</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common References</td>
<td>Common</td>
<td>Class Policy</td>
<td>Abstract Only, Concrete Only, Leaf Class, Standard</td>
<td>User</td>
</tr>
<tr>
<td>Common References</td>
<td>Common</td>
<td>Reviewer Role</td>
<td>Approver, Observer</td>
<td>User</td>
</tr>
</tbody>
</table>

**Related Topics**

- Lookups: 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). To access Manage Configurations, ensure that you have a role with the Setup Product Innovation privilege.

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
  - Approved manufacturer list status
  - Item attributes display settings
- 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.
Note:

- 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 Setup and Maintenance work area. Select the Product Management offering and Items functional 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 the Setup and Maintenance work area. Select the Product Management offering.

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.

**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.

**Configuring Access to Supplier Portal: Explained**

You can configure supplier users to perform the following from the Supplier Portal work area:

- View items and change orders created in Product Development.
  
  Provide users with the Supplier Product Designer role and the POS_ACCESS_SUPPLIER_PORTAL_OVERVIEW_PRIV privilege. Users can only view the items and changes orders that they are authorized to view.

- Generate item structure and change order reports.
  
  Provide users with the BI Administrator role and the EGO_GENERATE_ITEM_CHANGE_ORDER_REPORT_PRIV_OBI privilege.

Note:

- To avoid any errors while configuring access to supplier portal, ensure that the supplier user registration process has been completed successfully. Supplier user setup is handled in the Oracle Procurement Cloud.
- You can also create a copy of the role and assign the required privilege.

**Related Topics**

- Supplier Registration Process: Explained
- Supplier User Provisioning: How It Works
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,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pack Type, Primary Unit of Measure, Approval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Status, Organization, Lifecycle Phase, Item,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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

- 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.

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 (CDMCreateItem), 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 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>Functional Area</td>
<td>Tasks Enabled for Implementation</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------</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
14 Defining Workflow and Approval Management

Workflow and Approval Management: Overview

The BPM Worklist allows you to configure new item requests (NIRs), change orders, and change requests through a set of workflows. The following steps summarize the configuration:

- Run the Manage Task Configurations for Supply Chain Management task in the Product Management offering. This provides access to BPM Worklist.
- Update the workflow tasks by including details of deadlines and escalation policy.
- Select the notification type and provide additional details to configure the notification.
- Create approval groups that include a predefined set of users configured to act on the notification.
- Create approval rules that include conditions for approval.

Type of Approval

When configuring the workflow, you select the type of approval: user defined, rules-based, or web service based.

User defined approval allows you to add users or user groups as approvers, without specifying any conditions.

Rules-based approval allows you to create complex rules for managing the workflow. You use the BPM Worklist to create complex approval rules based on item, attributes, or attribute groups. For example:

\[
\]

Web service based approval allows you to define the list of change order approvers through a web service call. Note that the list of approvers must be defined before you change the approval status.

Example of a Change Order Workflow

Consider that manufacturing changes require review and approval before pricing changes are determined. You can create a change order workflow with multiple open and interim approval statuses and a final approval. Use the first set of open and interim approval workflows to review and approve the manufacturing changes. Use the second set to review and approve the pricing changes. You can then route the workflow for final approval.

Related Topics

- Change Order Approvals: Points to Consider
Setting up the BPM Workflow: Explained

Here is how you set up the new item request workflow. From the Setup and Maintenance work area, run the following tasks in the Product Management offering:

1. Run the **Manage New Item Request Type** task. Depending on the status type, you can do the following:
   - Enable or disable the request comment notification.
   - Setup automatic promotion or automatic demotion between statuses.
   - Select the entry and exit criteria for a status.

2. Run the **Manage Item Rule Set** task to create validation rules for the new item request.
3. Run the **Manage Item Classes** task to enable the new item request for an item class. In item management, include the definition workflow details such as steps, assignees, business or associated entities, and definition values.
4. Run the **Manage Approval Group** task to create an approval group responsible for the new item request approval.
5. Run the **Manage Task Configuration for Supply Chain Management** task. Select the participant header stage and create a business rule.

Here is how you set up the change order workflow. From the Setup and Maintenance work area, run the following tasks in the Product Management offering:

1. Run the **Manage Change Order Type** task. Depending on the status type, you can do the following:
   - Enable or disable the request comment notification.
   - Setup automatic promotion or automatic demotion between statuses.
   - Select the entry and exit criteria for a status.

2. Run the **Manage Item Rule Set** task to create validation rules for the change order.
3. Run the **Manage Approval Group** task to create an approval group responsible for the change order approval.
4. Run the **Manage Task Configuration for Supply Chain Management** task. Select the participant header stage and create a business rule.

Workflow Configuration in NIRs and Change Orders: Explained

In a new item request (NIR) the entire workflow is seeded. It contains the following statuses: Open > Definition > Approval > Scheduled > Completed.

Types of change orders: engineering change order (ECO), change order without revision control (NRCO), and commercialization change order (CCO). You create a change order by copying one of the change order types.

Here is the seeded workflow in engineering change order and change order without revision control: Open > Approval > Scheduled > Completed.

In the engineering change order, an approval is required and the revision value is incremented.

In a commercialization change order, the value of revision can be configured by the user. Here is the seeded workflow in the commercialization change order: Open > Scheduled > Completed.
You cannot change the sequence of the seeded status. However, some of the statuses can be configured.

When a change order type is created, the workflow is defaulted to the following statuses: Open > Scheduled > Completed.

For the purpose of auditing, you can create a workflow to process the change order without routing it through approval. However, if an approval is required, you can add an interim approval and an approval status. After the approval, the change order is automatically promoted to Scheduled. When the effective date is reached it is automatically promoted to Completed.

The figure shows the change order workflow being configured by using the Manage Change Order Type task.

### Open

When the NIR or change order is submitted to the Open status, a seeded request comment notification is sent. This informs the requester and assignee of the NIR or change order submission. Automatic promotion status can be set up for the open status as well as exit criteria. Because the Open status is editable, the Allow Update check box is automatically selected.

### Definition

Definition workflow is only available for NIRs. In the NIR type, you can configure automatic promotion and automatic demotion rules as well as entry and exit criteria. Definition workflow is defined for an item class; each item class can have its unique set of definition steps and assignees. Additional configuration must be completed using the Manage Item Classes task, in which tasks are defined per assignee or role. You can add multiple rows of the same step sequence number with different tasks for a parallel flow.
The figure shows the definition workflow being configured in an item class by using the Manage Item Classes task.

If a parallel definition workflow is configured with multiple rows having the same step number, users are notified (in parallel) according to the Response Required configuration. For the workflow to proceed, users defined for each row with the same step sequence number must complete their tasks. For example, if you set Response Required From as One and Assignee Type as Role, only one user with that role needs to complete the task. If you set it as All, all users with that role must complete the task.

If certain attributes require a value to be entered on item definition, you can select these attributes as required definition values.

Interim Approval and Approval

Interim approval can be used to progressively approve the item changes. There can be only one approval status in change order workflow. For example,

Open > Interim Approval > Open > Interim Approval > Approval > Scheduled > Completed

Interim approval is available only for change orders. To streamline the approval process, some of the configuration can be performed for a change order type.

Following features are available for both interim approval and approval:

- Automatic promotion and demotion.
- Entry and exit criteria.

You can allow updates only for interim approval. This enables you to update only the header attribute values to fulfill the entry and exit criteria. However item attributes cannot be updated during the interim approval even if the Allow Updates check box is selected (in the change order).

Note: Allowing updates affects the entry and exit criteria.
The figure shows approval configuration in a change order type. It is configured using the Manage Change Order Type task.

Header Stage Approval Configuration

Header stage assignment method can be rules based or user defined. You can set the Response Required From value to All or One. If it is set to All:

- Rejection from a single user rejects the item changes.
- Approval is required from all the approvers.

**Note:** Header stage approval cannot be disabled for NIRs and change orders.

Optional Approval Configuration

You can add optional approvers to the approval workflow. An optional approver is a person who interrupts the workflow only if a rejection is required. An approval from the optional approver is not considered as the final approval of the change order or NIR. However, a rejection immediately ends the approval workflow. If a header approver approves the workflow, then the approval task of the optional approver is automatically withdrawn. Note that the Response Required From option is not enabled for optional approval. The optional approval stage can be disabled. Notification for optional approval is sent as soon as the approval workflow begins.
Configuring Tasks Using BPM Worklist: Overview

Task configuration enables administrators to review and modify the approval rules defined by a workflow designer.

✏️ Note: Before you configure a task, click the Edit task icon.

Administrators can configure the following:

- **General**: Define basic information such as title, description, and priority. Selecting Hide task creator prevents the display of task creator’s name in the approval notification.
- **Data**: Review the message elements that compose the structure of the task payload. Do not edit this information.
- **Deadlines**: Specify the duration and expiration details.
  - **Due Date**: Indicates the date when the approval task is due for action. It is an indicator to remind the approver to respond by a certain time. The approver can also respond after the due date.
  - **Exclude Saturday and Sunday**: Excludes Saturday and Sunday while computing the due date, expiration date, and escalation date.
  - **Expiration Settings**: Specify the expiration policy at either task or assignee level, or skip it entirely. Includes the following:
    - **Escalate**: For example, if you use the escalation hierarchy configured in the user directory, the task can first be escalated to the user’s manager. If the user’s manager doesn’t take appropriate action within the specified duration, then the task is further escalated till it reaches the maximum escalation level or the highest approver. An escalated task remains in the user’s inbox even after the task has expired. However, the user is not allowed to take any action on that task.
    - **Access**: Configure the password policy for the task.
      - When the change order reaches the Approval status, users are prompted to enter their login password.

Configuring the Password Policy for Change Order Approval: Procedure

To configure the password policy:

1. In the Setup and Maintenance work area select the Product Management offering. Run the Manage Task Configurations for Supply Chain Management task.
2. Select ChangeOrderApprovalTask.
3. Click the Edit task icon.
4. Click Access and expand Actions.
5. Select the password policy.

When the change order reaches the Approval status, users are prompted to authenticate the process by entering their login credentials. The credentials are also used to audit the approval process.

If the change order is already in the Approval status when the password policy is enabled, the user does not receive the authentication prompt.
If the change order is in the Open status and the password policy is enabled, the user receives the authentication prompt when the change order reaches the Approval status.

Configuration Options in the BPM Worklist

Some of the key configuration options in the BPM Worklist are:

- Skip creator for Approval List: Prevents creators from approving their own task. You can also select Assign to Creator’s Manager to route the task to the manager.

  **Note:** If the routing is user defined, you cannot skip creators from approving their own task.

- Mandate Comments before updating these outcomes: Enforces reviewers to provide comments before approving or rejecting a task; might be useful for auditing.

- Notify these participants when error occurs: Allows certain administrators to receive notifications (in case of errors). If the task is approved by an assignee and the approval notification is again sent to the assignee, then a second approval is optional.

- Task Aggregation: Governs the routing of a task to different reviews in the approval cycle. If an approver is repeatedly assigned to a task, notifications can be aggregated and sent. If the task is approved by an assignee and the approval notification is again sent to the assignee, then the second approval is optional.

  Task aggregation includes the following options:
  
  - None: Do not aggregate notifications, send one per task assignment.
  - Once per task: For a given task, aggregate all notifications for a user and send a single notification for an assignee (during the task).
  - Once per stage: For a given stage, aggregate all notifications for a user and send a single notification for an assignee (during the stage).
  - All stages: Aggregate notifications for a user across all stages.

  **Note:** It is recommended to retain the default settings.

BPM Workflow Notifications: Overview

As the change order or new item request (NIR) progresses through the workflow, several notifications are sent out. Notifications inform participants about the occurrence of different events and enable them to take actions. These notifications are seeded and cannot be changed.

Individual notifications are sent for each row in the definition workflow steps.

If an assignee is present for multiple associations across items for a single step, then:

- The assignee receives notification for each task.
- Each task can be delegated as required.
• After the step sequence is completed, notification for the next step is sent.

In an approval notification, approvers can only view the lines assigned to them.

If the NIR or change order fails in a scheduled state, a consolidated notification is sent for all the failed lines.

The following notifications are sent: a detailed notification, and a task entry worklist table. The detailed notification displays the basic header attributes and assigned items. In an approval notification, an approver can enter comments and add attachments. The task entry notification is a one line entry in the worklist table of the item work area; you cannot open it in a separate window.

Configuring Workflow Notifications: Explained

The approval notification informs the respective approver to review the business transaction and take action. Notifications can be sent through email, or application bell notification.

Notification Mode

• All: Both email and bell notifications
• None: Neither email nor bell notifications
• Email: Only email and no bell notifications

Notifications indicate that a user or group is assigned with a task, or the task status has changed. Notifications are sent to different types of participants for different actions.

Managing email notifications includes the following:

• Enable the notification: Click the plus icon to add new notification entry, choose the task status, and notification recipient.
• Disable the notification: Select the appropriate action and click delete to disable the notification.

Users can also allow actions on notifications and enable attachments to be sent with notifications. Allowing actions enables recipients to approve or reject an object. If an attachment is present, users can send the attachment through email for review.

Notification Header

Users can add the company logo or company name in the notification header. The default value of notification headers is null.

Any new notification added by the user includes the following header value:

```html
"concat(string('Task '), /task:task/task:title, string(' requires your attention.'))"
```

It is recommended to change the header value to null.

To change the company logo, include your company URL and provide a proper alternative text.

For example, img src="http://b-i.forbesimg.com/joshbersin/files/2013/07/company-logo3.jpg" width="230" height="69" alt='Company Logo'

Bell Notification Sync-up

The Synchronize Bell Notifications ESS job is used to sync the Bell Notification dialog box with online BPM notifications. All the completed notifications are removed from Bell Notifications.
The FND_MANAGE_SCHEDULED_JOB_DEFINITION_PRIV privilege, is required to schedule or execute the Synchronize Bell Notifications ESS job.

**Managing Bell Notifications**

Enable/Disable Bell Notification: Uses the settings configured for email notifications. If the email notification is disabled for specific action, the user does not receive bell or email notification.

Enable Reminder: Send task reminders based on the time when the task was assigned to a user or the time of task expiration. The number of reminders and the interval between the reminders can also be configured.

More Options: The application provides additional options to configure email notification.

- Make notification secure (exclude details): Prevents any business transaction details from appearing in email notifications.
- Do not send multiple notifications for the same task.
- Hide End User Web URL in notifications: Removes the link that provides access to the task in Workspace Application instead of the underlying transaction.
- Make notification actionable: Controls the following links in notification email: Approve, Reject, and Request More Information.
- Send task attachments with email notifications: Allows supporting documents to be attached to email notification.

**Task Assignment and Routing: Explained**

The BPM Worklist supports declarative assignment and routing of tasks to a single user or group. Additionally, pattern-based support is available for scenarios that require detailed task assignment and routing.

**Stage**

A stage is a way of organizing the approval process for blocks of participant types. In the Product Development and Product Hub work areas, there is only one stage and it's in a parallel mode. In the parallel mode, the task gets assigned and notifications are sent to all participants at the same time. Within each stage, you can have one or more blocks of participant types. It is recommended not to modify the default settings in stage.

**Participant**

A participant is a user or set of users in the assignment and routing policy definition.

**Participant Type**

A participant type corresponds to a user or group. The workflow supports declarative patterns for common routing scenarios such as management chain, and group vote. The following participant types are available:

- Single approver: The participant maps to a user or group.
  
  For example, a vacation request is assigned to a manager. The manager must act on the request three days before the vacation starts. If the manager formally approves or rejects the request, the employee is notified about the decision. If the manager does not act on the request, the request is treated as rejected and actions are similar to a formal rejection.

- Parallel: The participant indicates that a set of people must work in parallel. This pattern is commonly used for voting. For example, multiple users in a hiring scenario must vote to hire or reject an applicant. You specify the voting percentage that is needed for the outcome to take effect, such as a majority vote or unanimous vote.
• FYI (For Your Information): Corresponds to a single user, group, or role. This pattern indicates that the participant just receives a notification task and the business process does not wait for the participant’s response. Participants cannot directly impact the outcome of a task, however in some cases can provide comments or add attachments.

For example, a regional sales office is notified that a product has been approved by the regional manager and is being passed on to the state wide manager for approval or rejection. FYI participants cannot directly impact the product approval. They can only provide comments or add attachments.

Approval Groups: Explained

An approval group consists of a static and predefined set of users configured to act on a task. Approval tasks are routed to an approval group in parallel mode. For example, you can create an approval group called Line Managers comprised of users from the finance department who need to participate in approving a task.

New approval groups can be created, or existing approval groups can be modified using the BPM Worklist. The approval group can then be selected in the list builder.

If the Allow empty groups attribute is set to True and the approval group doesn’t contain any member, then the rule evaluation progresses without displaying any error. If the rule evaluation does not result in a valid assignment, the BPEL process manages the scenario. In such scenarios, the transaction is either rejected or moved back to the original status. It is recommended to set the Allow empty groups attribute to False.

Task, Stage and Participant

The BPM Worklist supports the Change Order Header Stage.

The figure shows the stages and participant within a task

![Diagram of stages and participant](image)

Note:

• The following stages are not supported: Change Order Line Stage, Change Order Line Serial Stage, and Change Order Line Parallel Stage. By default, they are set to Ignore Stage.

• Though you see the line stage approval configuration in the BPM workflow configuration screen, it is recommended not to make any changes in the line stage approval configuration.

The seeded approval tasks, stages, and participants in change order and new item request:

Change Order Approval Task

• Change Order Optional Approval Stage
  o Change Order Optional Approvers
• Change Order Header Approval Stage
  o Change Order Header Rule Based Parallel Approvers
  o Change Order Header Rule Based Single Approver
  o Change Order Header User Defined Parallel Approvers
  o Change Order Header User Defined Single Approver

New Item Request Approval Task

• New Item Request Optional Approval Stage
  o New Item Request Optional Approvers

• New Item Request Header Approval Stage
  o New Item Request Header Rule Based Parallel Approvers
  o New Item Request Header Rule Based Single Approver
  o New Item Request Header User Defined Parallel Approvers
  o New Item Request Header User Defined Single Approver

Approval Management: Explained

Transaction reviewers are provided with the following:

• Bell icon
• BPM Worklist
• Email notification

Approvers are provided with the following actions:

• Approve: Once a task is sent for approval, the approver can use the approve action to continue the task.
• Reject: Task assignee can reject the task to prevent further approval.
• Request More Information: If the task assignee wants more clarification from the requester, the task can again be sent to the requester. Users can also specify who requested the information. The task is not rejected and once the required information is provided, the approval flow begins again.
• Reassign: The task assignee can send the task to another user for approval. The new user’s hierarchy is used for approval. For example, if a user thinks that the task is relevant to another department, he can reassign it.
• Delegate: A user can delegate the task to another user. After approval by the delegate, the initial user’s hierarchy is used for approval. The delegate can still act on the task after task expiry.
• Withdraw: The task initiator can withdraw the task after the approval has been initiated.
• Escalate: A user can escalate the task from the current assignee to the supervisor.
• Claim: A task that is assigned to a group or multiple users must first be claimed. Claim is the only action available in the task action list for group or multi-user assignments. After the task is claimed, all applicable actions are listed. Claim is available only if auto claim is disabled and the response required from is set to One.
• Dismiss: Used for a task that requires the person (acting on the task) to acknowledge the receipt. This is similar to an FYI notification, which does not involve any action.
• Resume: A task that was halted by a Suspend action can be worked on again.
• Release: Releasing a claimed task makes it available to other assignees. A task assigned to a group or multiple users
  can then be claimed by the other assignees.
• Suspend: The expiration date remains suspended until the task is resumed. After suspension, options to update a
  task are disabled.

**Note:** The Suspend and Resume tasks are only available for users with the BPMWorkflowSuspend role.

## Entry and Exit Criteria: Explained

Rules based entry and exit criteria can be used to add an additional layer of validation to the change order or new item
request (NIR). The criteria can be selected for different workflow statuses in the change order or NIR.

The entry criteria is available for the following statuses: definition, interim approval, and approval.

The exit criteria is available for the following statuses: open, definition, and interim approval.

You can create validation rules using the **Manage Item Rule Set** task; set the association type to change order or NIR. Prior
to creating the entry and exit criteria, create a change order type and set it to rules based approval.

The following attributes are supported when creating the criteria:

**Change Header Main**

• Priority
• Reason
• Need-by Date
• Requested By
• Description
• Descriptive Flexfield

**Change Line**

The change line entity does not support line attributes. You can however create the criteria using descriptive flexfield values
from change line, for which you need to specify the entity. Example:

```
[ChangeHeader].Flexfield[CHHD_Glob_Seg_Char_1]
```

Example of the exit criteria for open status: Consider that the priority code of the change order is high. Then the need-by date
 can be set as mandatory for the change order to be promoted to the next status in the workflow.

```
If Expression
[ChangeHeader].[ChangeHeaderMain].[PriorityCode] == "HIGH"

Validation Condition
isNull([ChangeHeader].[Change Header Main].[Need-by Date]) == false
```

Example of the entry criteria for approval status: If the value in the Reason field is Cost, then priority of the change order is set
to High.

```
If Expression
[ChangeHeader].[ChangeHeaderMain].[ReasonCode] == "Cost"

Validation Condition
```
[ChangeHeader].[ChangeHeaderMain].[PriorityCode] == "HIGH"

Creating an Approval Rule: Worked Example

Approval rules in change orders and new item requests (NIRs) can be based on items, attributes, or attribute groups. You can create rules to route each attribute group to a different approver or approval group.

Rules are executed based on:

- The new item data including the unchanged data from production or source version and
- The modified data.

Prerequisites

- Ensure that EFFs are deployed and synchronized in BPM.
- For the change order type, ensure that the approval is set to rules-based.

To do this, run the Manage Change Order Types task in the Product Management Offering. Edit the change order type to set the approval to rules-based.

This table summarizes key decisions to consider, and the decisions made when creating an approval rule for a change order type.

<table>
<thead>
<tr>
<th>Decisions to Consider</th>
<th>In This Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who are the users responsible for approving the change order?</td>
<td>Create an approval group.</td>
</tr>
<tr>
<td>Have you created an approval rule for the change order type?</td>
<td>Create an approval rule.</td>
</tr>
</tbody>
</table>

The figure shows an approval group and its members.

![Approval Group and Members](image-url)
The figure shows one of the steps in creating an approval rule.

Create a change order approval group

1. From the Setup and Maintenance work area, run the **Manage Approval Group** task in the Product Management offering.
2. Click **Create Approval Group**.
3. Enter a name for the group and click **Save**. For example, create a PricingApproval group to route all pricing changes to this group.
4. Click **Add Member**.
5. In the **Add to Group** dialog, select either a user or an existing approval group and click **OK**.
6. Save the approval group.

Create a change order approval rule and include the approver

1. From the Setup and Maintenance work area, run the **Manage Task Configurations for Supply Chain Management** task in the Product Management offering. Click the task under the relevant functional area.
2. In the BPM Worklist page, search for ChangeOrderApprovalTask and select that task. Note that ChangeOrderApprovalTask can also be used to create an approval rule for change request.
3. Click **Assignees**. The Change Order workflow appears.
4. Click the **Pencil** icon next to the **Save** icon.
5. In **Change Order Header Rule Based Approver**, click to dot icon and select **Go to rule**.
6. In Rulesets, click the **Plus (Add Rule)** icon.
7. Enter a name for the rule and click **Expand** to add the rule.
8. Select the condition when the rule is required to be executed, and the outcome of the rule. For example, to control the approval of all change orders created by a user, you can route them to a particular approval group.
   - If you create a rule based on the change order priority, the name and case of the priority must match with the name and case of the lookup code in the Manage Change Priorities task.
   - If you create a rule based on the item class, ensure that you use the internal name mentioned in the item class.
9. Click **Save**.
10. To activate the rule, click **Commit task**.
Routing Items to a Particular Group: Sample Rule

You can create rules using the BPM Worklist.

Sample Rule

If an item belonging to a particular class exists in the change order and the priority of the change order is high, then the approval can be routed to the members belonging to a group. Use the advanced mode and tree mode to create the following rule:

```
Select SoaOLabel.ChangeOrderHeaderRuleBasedSingleApprover
Select ChangeOrderHeaderRuleSet1
ChangeOrderHeaderRuleSet1:
View: IF/THEN Rules
Rule:
ROOT: ChangeObjectApproval
IF
Change_Order is a ChangeObjectApproval
And
Lists is Lists
And
{
RevisedItemLine is a Change_Order/revisedItemLineApproval and
Change_Order/revisedItemLineApproval RL.contains RevisedItemLine.item.primaryUomCode and
RevisedItemLine.item.primaryUomCode is "Box"
}
Then
List builder = Approval Group
Response Type = Required
Approval Group = "ChangeLineApprovalGroup"
Allow empty groups = False
```

FAQs for Workflow and Approval Management

What happens if I start synchronization?

Synchronization affects EFF metadata of all users in the environment. It synchronizes all EFF metadata available from 1 month, 10 days and 3 hours.

Why am I prompted to start synchronization even after I have synchronized the data?

Another user might have modified EFFs, or the session might have timed out before the synchronization is complete.
How long does the EFF synchronization process take?

Depending on the environment set up and EFF configuration, synchronization might require between 5 to 45 minutes.

Why did the workflow stop responding?

The workflow might stop responding due to one of the following reasons:

- Approval rule encounters an empty approval group.
- Approval rule not being associated with an approval group.
- One or more change order lines doesn’t match any rules in the approval task.
- EFFs not being generated.
- Inactive rules.
- Rules unavailable for the header stage.
- In a new item request, if a reject rule states that Approval Status is Approved and Description = ABC, then the new item request is rejected. Approval notification is not sent.
- A new item request definition step assigned to a role and the role not being assigned to any user.
- An error in SOA workflow. Use Enterprise Manager to view workflow errors.
- Use of an incorrect case to specify the value of a parameter such as priority code.

Why is the advanced mode required while defining rules in the header stage?

Because the NewItemLineApproval object is available only through the ChangeObjectApproval object.

Can I create dynamic approval groups for a line and for on-premise customers?

Dynamic approval groups cannot be created for a line or for on-premise customers.

How can I indicate that a workflow notification is opened by another user?

You can expose the Claimed by value on worklist notifications. The claim action is only available for tasks having a single participant, in which an assignee can act on behalf of all users receiving the notification. It is controlled using the Manage Task Configuration for Supply Chain Management task. Select the enable Auto claim option.
Why did the change order get struck in the workflow in spite of having optional approvers?

A rejection from an optional approver and absence of a mandatory approver might cause the change order to get struck in the workflow. It is important to define mandatory approvers.

What's the difference between reassign and delegate?

A user can reassign the task to another user for approval. The new user’s hierarchy is used for approval. For example, if a user receives an approval request relevant to another team, he can reassign it to a user in another team.

A user can delegate the task to another user. After approval by the delegate, the initial user’s hierarchy is used for approval. The delegate can still act on the task after task has expired.

Can I specify rules at different levels when using dynamic approval groups for change orders?

For on-premise customers, rules can be specified at both header and line levels.

How can I use BPM Worklist to configure the NIR or change order for autoapproval?

When creating the approval rule, set Auto Action Enabled to True, and Auto Action to Approve.
Glossary

**attribute**
A named entity whose value describes a product item. Attributes can be organized into attribute groups. You can search for items based on attribute values, by adding attribute fields when using Advanced Search (but you cannot search on transactional attributes). You can compare the attribute values of selected items returned by an item search.

**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.

**database resource**
An applications data object at the instance, instance set, or global level, which is secured by data security policies.

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

**ESS**
Acronym for Enterprise Storage Server. An application that optimizes data storage.

**extensible flexfield**
Expandable fields that you can use to capture multiple sets of information in a context or in multiple contexts. Some extensible flexfields let you group contexts into categories.
flexfield
A flexible data field that you can configure such that it contains one or more segments or stores additional information. Each segment has a value and a meaning.

flexfield segment
An extensible data field that represents an attribute and captures a value corresponding to a predefined, single extension column in the database. A segment appears globally or based on a context of other captured information.

GTIN
Abbreviation for Global Trade Identification Number

job role
A role, such as an accounts payable manager or application implementation consultant, that usually identifies and aggregates the duties or responsibilities that make up the job.

role provisioning
The automatic or manual allocation of a role to a user.

structure
A bill of materials. A structure contains information on the parent item, components, attachments, and descriptive elements.

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.